今使ってる道具も、今やってる仕事も、みんな誰かのものづくりによって生まれた発明品。

ISSN 2187-9664

for the sophisticated shell scripters

シェルスクリプトに何でもおまかせ
Open usp Tukubai シンプルレシピWeb編

デール・ダハティー氏に訊く、技術者哲学

TechLION再録
砂原秀樹
次の世代とともに進む未来

好評連載
原色Linux美女図鑑
菅雄一 中小企業手作りIT化奮戦記
新連載・長谷川猛 うにっくすなやつら

開催10年目 LL Decadeイベントレポート

From Editor

今年も秋がやってきました。読書の秋、スポーツの秋。多くの作物も収穫され、クリエイティブな季節ですね。

そんな秋にふさわしく、今回の USP MAGAZINE の裏テーマはものづくりです。

技術者哲学の記事でインタビューに伺ったのは、ものづくりコミュニティー雑誌「Make」発起人のデール・ダハティー氏。オライリー・メディアの創業から UNIX 書の執筆、そして雑誌 Make の発行に至るまで、バラエティー豊かなものづくりの精神が氏から伝わってきました。

一方で、今回の TechLION 再録記事に登場している砂原秀樹先生がまた、ものづくり精神あふれる方だったのです。日本のインターネットを作り、新しい学科を作り、そして今、次世代を担う人材を育てていらっしゃいます。

TechLION のトーク中、自らを「立ち上げ屋」と称していらっしゃいました。あのようにしてプロジェクトを次々と成功させてきた秘訣はなんだろう……、その一つはトークにも滲み出ていました。それは見識と冗談を絶妙なバランス感覚で繰り出し、各界の人々と渡り合う能力なのでしょう。

だから今回の TechLION 記事。当時の先生の雰囲気や人柄がありのまま伝わるよう、普通の雑誌なら割愛されてしまうであろう冗談や口調を、普段にも増して残しました。

ものづくりに長ける人は仲間づくりも長けている。二つの記事を編集するなかで、そんな共通点が見えてきました。

秋と言うクリエイティブな季節、皆さんもスキルを磨くと共に勉強会などに参加し、仲間を増やしてみてはいかがでしょう。

USP MAGAZINE 編集部

Contents

特集 1　シェルスクリプトを本気で使い倒す Open usp Tukubai シンプルレシピ Part2 … 3

ユニケージエンジニアの作法　第 4 回 …… -

今私たちは何を学ぶべきか　第 6 回　大岩元 …… -

TechLION 再録　砂原秀樹 ―― 思えば遠くに来たもんだ。次の世代とともに進む未来 …… -

新連載　うにっくすなやつら　長谷川猛 …… -

原色 Linux 美女図鑑 ―― 第 5 回　西本恭子 …… 11

LL DECADE イベントレポート …… 12

漢の UNIX　第 3 回　後藤大地 …… -

やわらかマッドサイエンティスツのプログラミング講座 ～ ジャクソン構造図編 …… -

特集 2　デール・ダハティ氏に訊く、技術者哲学 …… 16

中小企業手作り I T 化奮闘記　第 4 回　菅雄一 …… -

シェルスクリプト大喜利　第 6 回 …… 20

Tech 数独 …… 24

天地概況　奈須蛍路 ／ 編集後記 …… 25

※ 薄字で記している記事は、よりぬき版には収録されておりません。また、本誌とは収録順序が若干異なります。

特集1

シェルスクリプトを本気で使い倒す

Open usp Tukubai シンプルレシピ

Part2・Web編

USPマガジン編集部

Tukubai、
それはシェルスクリプトをプログラミング開発言語として
本気で使うためのコマンドセット。

「本気で使う」ということは、決して無理矢理使うことではない。
実用的に使うことができるという意味である。

前回は帳票作成と数理演算への活用例を示した。
今回はWebアプリケーション開発への活用例を示そう。

レシピの中で紹介されているコードのいずれもが、
簡潔な記述で収まっていることに驚くことだろう。
それこそが本気で使えることの証なのだ。

準備
ShellScriptで**Web**アプリ開発。
それも**Tukubai**の得意技。

レシピ1
HTMLで書かれたデータを取り込む。

レシピ2
HTMLを動的に作成する。

レシピ3
Ajaxで**Web**フォームの使い勝手をよくする。

オマケ
宝くじチェックアプリを使ってみる。

準備　シェルスクリプトでWebアプリ開発。それもTukubaiの得意技。

Tukubai は、シェルスクリプトをプログラム開発言語として本気で使うべく作られたコマンド集だ。これは前号の本特集でも述べた。

しかし、Tukubai の説明をしている中でよく返される質問がある。「なるほど、HTML を担当する Java や Perl や Ruby 等と連携して使うのですよね？」というものだ。

連携が可能か否かという意図かもしれず、とりあえず「はい」と答えるが、Tukubai はそうやって連携させて使う目的で作られたものではないことをいつも必ず付け加えている。**Tukubai の本来の使い方は Web インターフェースも含めて丸ごとシェルスクリプト**で作ってしまうやり方なのだ。

シェルスクリプトで Web 周りまで実装するなどというと、笑い話に聞こえて受け入れ難いかもしれない。だが、それを実用レベルにまで引き上げる存在こそが Tukubai なのだ。本特集の最後では流行りの Ajax まで扱っているが、冒頭に述べた「開発言語としての本気っぷり」をとくとご覧いただきたい。

シェルスクリプトを使うのは意地ではない、実用的だから

もう一つ、よく受ける質問として「できるのはわかりますけど、実用的じゃないですよね？」というのがある。そんなことはない。理由として、次の二つを挙げたい。

MVC の全てをシェルで発想できる

Web アプリ開発でよく用いられる概念で MVC(Model: データアクセス、View : 表示、Controller:UI からの入力) というものがある。

Tukubai を用いた開発ならば、これら全てを馴染み深いシェルスクリプト一つで発想することができる。これは、開発環境をコンパクトにし、かつ見通しをよくすることにもなる。

コードの短さが、開発を迅速化

MVC という発想が生まれた理由の一つは、アプリの構造をこの 3 つに分離して部品化し、開発を迅速化することにあったはずだ。

本特集で出てくるコード見て欲しい。短いのだ。Unix・Tukubai のコマンドが持つ機能やその粒度がよく考えられ、適切だからなのだが、短いコードは開発の迅速化に寄与する。

Open usp Tukubai をインストールする

インストールは簡単。アーカイブをダウンロードして解凍し、make install するだけ。前号を見てインストールしたという方も、その後更新された（本誌発行時点で 2012 年 8 月 2 日版）ため、最新版に更新してもらいたい。本号のレシピでは、そこで追加されたコマンドも利用している。

ではインストール方法を詳しく見ていこう。

事前に必要なもの

◎ Python（2.4 以降の 2.x 系列）

Open 版である Open usp Tukubai のコマンドは Python（2.4 以上の 2.x 系用）で出来ている。よって事前にインストールしておこう。Python の導入についての詳細は、Python サイト（http://www.python.org/）や、日本 Python ユーザ会さんのサイト（http://www.python.jp/）を参照して欲しい。

Open usp Tukubai の導入

Open usp Tukubai のダウンロードは Tukubai に関するポータルサイトである「UEC」（usp engineers' community、https://uec.usp-lab.com/、→**写真 1**）にて可能である。

このサイトの上部メニューから、

→ "Tukubai"

→ "Open usp Tukubai ダウンロード "

と進むと、そこで最新版が公開されている。wget や fetch コマンド等を使って取得した後、次のようにして解凍、およびインストールする。

```
> tar jxf open-usp-ukubai-yyyymmdd.tar
.bz2
> sudo make install
```

レンタルサーバー等、root 権限が与えられておらず、ホームディレクトリー等にインストール先を変更したい場合、次のように make に与える LOCALBASE 変数でそのディレクトリー指定できる。

写真 1 . Tukubai のポータルサイト"UEC"

```
> tar jxf open-usp-ukubai-yyyymmdd.tar
.bz2
> make LOCALBASE=/home/USERNAME/tukuba
i install
```

この場合は、環境変数 $PATH に Tukubai コマンドのパスを追加すること。上記例の場合、/home/USERNAME/tukubai/bin となる。

尚、アンインストールする場合は make install の代わりに make deinstall とすればよい。

FreeBSD での導入は更に簡単！

FreeBSD では ports コレクションに Open usp Tukubai が登録されている。従って、Open usp Tukubai アーカイブファイルのダウンロード作業も、事前の Python インストール作業も不要だ (自動的に行われる)。

具体的には root になって下記のコマンドを打ち込むだけである。

```
# cd /usr/ports/devel/open-usp-tukubai
# make install clean
```

記事中のプログラムはUECサイトにてダウンロード可

これで準備完了。シェルスクリプトによる Web アプリの作られ方をとくと見て欲しい。

前回同様、掲載レシピのプログラム、及びサンプルデータを、先程紹介したポータルサイト UEC にてダウンロードできるようにしてある。是非併せて活用してもらいたい。

HTMLで書かれたデータを取り込む。

問題

みずほ銀行の宝くじ公式ページに載っている当せん番号一覧を表にして保存したい。

ただ元データは HTML になっていてタグがいっぱいついている。これを外し、プレーンテキストの表にできないだろうか。

解答

まず cURL(curl コマンド) の力を借りる。これで目的の URL から生の HTML テキストを得られる。あとはこれを、sed や AWK 等を使って地道に整形していけば完成だ。

整形のコツは、一旦 HTML テキストの中にある改行を全部取り去って、必要なタグの前（または前後）にだけ改行を入れることだ。

サンプルプログラム

宝くじの公式ページ（**写真１**）内の「ジャンボ宝くじ」の当せん番号を収集するプログラムを作成した（**リスト１**）。ただし当せん番号案内のトップは各抽せん回へのリンクになっており、実際の番号はそのリンク先にあったため、トップページでリンクを収集した後、さらに各々のページへアクセスし、当せん番号を収集している。

宝くじには「前後賞」や「組違い賞」といった変則的なデータがあるので、それらについてもきちんと展開するようにしてある。最終的に第一フィールドから、抽せん日、宝くじ名、等、賞金、組、番号というテキスト表データ（LOTTERY_DAT）にした。

画面１．宝くじ公式ページ＠みずほ銀行（2012 年 9 月現在）

リスト１．LOTTERY.DB：「ジャンボ宝くじ」当せん番号案内一覧を収集するプログラム

```
 1 #!/bin/sh
 2 #
 3 ##########################################################################
 4 # LOTTERY.DB 当たりたいなら買うしかない。宝くじデータベース！！
 5 # (SHELLというディレクトリを作成し、その下に配置してください)
 6 #
 7 # Written by N.Tounaka(tounaka@usp-lab.com) / Date : 13 Aug.2012
 8 # Arrenged by USP MAGAZINE(mag@uap-lab.com) / Date : 08 Sep.2012
 9 ##########################################################################
10
11 # シェル変数のセット
12 homedir=$(pwd); homedir=${homedir%/*}
13 pomd=$homedir/POMPA
14 top="http://www.mizuhobank.co.jp"
15 LF=$(printf '¥¥¥012_');LF=${LF%_}
16
17 # tacコマンドの定義(*BSDのみ)
18 [ -n "$(uname | grep BSD)" ] && alias tac='tail -r'
19
20 # 1.何はともあれ、宝くじ当せんデータベース(テキスト表)をつくります
21 # 宝くじ公式ページに記されているジャンボ宝くじの一覧を取り出す
22 curl -s ${top}/takarakuji/tsujyo/jumbo/index.html                |
23 # 必要な区間の行(<table>〜</table>を抽出                          #
24 sed -n '/<table/,/<¥/table>/p'                                    |
25 # 一旦改行コードを全部取り去ってしまう                            #
26 tr -d '¥n'                                                        |
27 # <a>タグの処理: リンクURLとリンク文字列だけ抜き出す              #
28 sed 's!<a [^>]*href="¥([^"]*¥)"[^>]*>¥([^<]*¥)</a>!¥1 ¥2!g'       |
29 # <th>タグを<td>と同等に扱うため、置換                            #
30 sed 's!<th!<td!g;s!</th!</td!g'                                   |
31 # <tr>タグが出てきたらその手前に改行コードを挿入                  #
32 sed 's/<tr/'"$LF"'<tr/g'                                          |
33 # <td>タグが出てきたらタグを外し,印"@"を付け,前後に改行を挿入     #
34 sed 's!<td[^>]*>¥([^<]*¥)</td>!'"$LF"'@¥1'"$LF"'!g'               |
35 # <td>タグ要素はスペース区切り,<tr>タグでは改行し,テキスト表に    #
36 awk '/^<tr/{if(time>0){print ""}else{printf(" ")}time++}          ¥
37      /^@/  {printf("%s ",$0)}                                      ¥
38      END   {print ""}'                                             |
39 # <td>タグ要素のうち、空だったものを "_" に変換する               #
40 sed 's/ @ / _ /g'                                                 |
41 # <td>タグ要素に付けていた印"@"を完全に除去                       #
42 tr -d '@'                                                         |
43 # テーブルの項目名(元<th>要素)の行を取り除く                      #
44 tail -n +2                                                        |
45 # "yyyy年m月d日"→"yyyymmdd"                                       #
46 sed 's/^¥([0-9]*¥)年¥([0-9]*¥)月¥([0-9]*¥)日/¥1,0¥2,0¥3/'         |
47 sed 's/^¥([0-9]*¥),0*¥([0-9][0-9]¥),0*¥([0-9][0-9]¥)/¥1¥2¥3/'     |
48 # 抽せん日、ＵＲＬ、宝くじ名を順番に取り出し、                    #
49 while read day url name; do                                         #
50   # 当せん番号を調べる                                              #
51   curl -s ${top}${url}                                              |
52   # 先程と同様に、HTML中のテーブル要素をテキスト表に変換            #
53   sed -n '/<table/,/<¥/table>/p'                                    |
54   tr -d '¥n'                                                        |
55   sed 's!<th!<td!g;s!</th!</td!g'                                   |
56   sed 's/<tr/'"$LF"'<tr/g'                                          |
57   sed 's!<td[^>]*>¥([^<]*¥)</td>!'"$LF"'@¥1'"$LF"'!g'               |
58   awk '/^<tr/{if(time>0){print ""}else{printf(" ")}time++}          ¥
59        /^@/  {printf("%s ",$0)}                                      ¥
60        END   {print ""}'                                             |
61   sed 's/ @ / _ /g'                                                 |
62   tr -d '@'                                                         |
63   #   なる文字列を含む場合があるのでコイツを空白に変換         #
64   sed 's/ / /g'                                                |
65   # １等前後賞のためのレコード追加                                  #
66   tac
```

（注釈）

- sed コマンドで改行への置換を綺麗に書くため、シェル変数に改行をセットしている
- BSD 系 OS には tac コマンドが無いが、相当の機能はあるためエイリアス定義している
- curl コマンド（指定 URL の HTML データを標準出力に流す）
- まず sed を使い、今回必要な <table>~</table> 区間のみを抽出
- HTML 内の改行を一旦全消去
- <td>~</td> はタグを外し先頭にプレフィックス "@" を付けている
- 一旦改行を全消去後、必要なタグにだけ付け、AWK コマンドでタグに応じた処理をしている
- <td>~</td> の中が空だった場合への対処（"_" としている）
- 日付フォーマット変換 (YYYYMMDD)
- 当せん番号一覧の記載されいるページについても同様の方法で、タグをパースしている

次ページへ続く→

このシェルスクリプトによって **Text1** のような当せん番号テーブルデータが生成される。

```
67   awk ' {if($1=="1等の前後賞"){s=$1;m=$2}                         ¥
68         else if($1=="1等")                                        ¥
69           {print;print s,m,$3,$4+1"番";print s,m,$3,$4-1"番"}     ¥
70         else print                                             }' |
71   # 1等組違い賞のためのレコード追加                                #
72   awk ' {if($1=="1等の組違い賞"){s=$1;m=$2}                       ¥
73         else if($1=="1等"){print;print s,m,"各組共通",$4}         ¥
74         else print}'                                              |
75   awk 'NF==4'                                                     |
76   sort -n -k1,1                                                   |
77   awk '{print "'$day' '$name' ",$0}'                              ;
78 # LOTTERY_DAT 1:抽せん日 2:宝くじ名 3:等 4:賞金 5:組 6:番号    #--
79 done                                              > $pomd/LOTTERY_DAT
80
81 # 2.抽せん日選択のドロップダウンリスト生成用に、インデックスを作る
82 self 2 $pomd/LOTTERY_DAT                |
83 uniq                                    |
84 # KIND_LIST    1:インデックス 2:宝くじ名
85 awk '{printf("%02d %s¥n", NR, $1)}' > $pomd/KIND_LIST
86
87 # 3.組番号選択のドロップダウンリスト生成用に、インデックスを作る
88 seq 99                                  |
89 awk '{printf("%02d¥n",$1)}'             |
90 # GROUP_LIST   1:インデックス 2:組
91 self 1 1                                > $pomd/GROUP_LIST
92
93 exit 0
```

前後賞や組違い賞の行があったら、番号と金額を適宜直した行を複製する

これらのデータは、レシピ2で動的HTMLの生成に利用する

解説

実はこのプログラムには Tukubai コマンドが一つも出てこないのだが、Web アプリ開発で HTML や XML のパースは欠かせぬ話ということで取り上げた。

大抵の言語では DOM や SAX を使うところ、シェルスクリプトなら sed や AWK、tr といったテキストフィルターコマンドだけで軽くこなすこともできる。

改行を入れ直す

シェルスクリプトで **HTML や XML のタグをパースするコツは、改行を一旦全部除去してから必要なところにだけ再び挿入**することである。

例えば HTML には <table> というタグがある。言うまでも無くこれは HTML 上でテーブルを表現するためのタグである。この中には <tr> や <td> といった行、列を示すタグがある。このテキストから行、列を正しく認識してテーブルの中身を取り出すにはどうすればよいか。

まず sed で <tr> の前後に改行を入れ、<tr> を単独行にしておけばよい。その後 AWK で "<tr" 始まり行を読み込んだ時に、新しいレコードに移ったと認識できるからだ。

<td> に関しても、<td> ～ </td> という単位で単独行にしておけば、やはり "<td" 始まりの行を一つの列であると認識して、値を収集できる。

この時 <td> から </td> の間に改行があると都合が悪いので、最初に予め取り去っておくというわけだ。幸い HTML は、
 を改行として扱っているから問題にならない。

不正な HTML でもへっちゃら

HTML ともなれば、タグの間違いがまかり通るのは日常茶飯事。なんと執筆時点 (2012 年 9 月) で、みずほ銀行のジャンボ宝くじ当せん番号案内トップページの HTML データも少々間違っている。抽せん回一覧ページで、抽せん日の <td> が </th> で閉じられている。

DOM や SAX といった外部のパーサーに頼っていた場合、こういうデータの処理で苦労するかもしれないが、こちらは自前で作っているので、さほど悩むこともなく対応してしまえるというわけだ。

Text1. LOTTERY.DB が収集した、ジャンボ宝くじの当せん番号一覧テーブルデータ

```
20120807 第625回全国自治宝くじ(2000万サマー) 1等 2000万円 組下1ケタ0組 120302番
20120807 第625回全国自治宝くじ(2000万サマー) 1等 2000万円 組下1ケタ0組 139302番
20120807 第625回全国自治宝くじ(2000万サマー) 1等 2000万円 組下1ケタ1組 101210番
20120807 第625回全国自治宝くじ(2000万サマー) 1等 2000万円 組下1ケタ2組 136935番
20120807 第625回全国自治宝くじ(2000万サマー) 1等 2000万円 組下1ケタ8組 146564番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ0組 120301番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ0組 120303番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ0組 139301番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ0組 139303番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ1組 101209番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ1組 101211番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ2組 136934番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ2組 136936番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ8組 146563番
20120807 第625回全国自治宝くじ(2000万サマー) 1等の前後賞 10万円 組下1ケタ8組 146565番
20120807 第625回全国自治宝くじ(2000万サマー) 2等 1万円 下3ケタ 871番
20120807 第625回全国自治宝くじ(2000万サマー) 3等 300円 下1ケタ 5番
20120807 第624回全国自治宝くじ(サマージャンボ) サマーバケーション賞 10万円 下4ケタ 1211番
20120807 第624回全国自治宝くじ(サマージャンボ) 1等 4億円 41組 105615番
20120807 第624回全国自治宝くじ(サマージャンボ) 1等の前後賞 5000万円 41組 105614番
20120807 第624回全国自治宝くじ(サマージャンボ) 1等の前後賞 5000万円 41組 105616番
20120807 第624回全国自治宝くじ(サマージャンボ) 1等の組違い賞 10万円 各組共通 105615番
20120807 第624回全国自治宝くじ(サマージャンボ) 2等 500万円 68組 185568番
20120807 第624回全国自治宝くじ(サマージャンボ) 2等 500万円 77組 162668番
20120807 第624回全国自治宝くじ(サマージャンボ) 3等 100万円 各組共通 189984番
20120807 第624回全国自治宝くじ(サマージャンボ) 4等 1万円 下3ケタ 546番
20120807 第624回全国自治宝くじ(サマージャンボ) 5等 3000円 下2ケタ 36番
20120807 第624回全国自治宝くじ(サマージャンボ) 6等 300円 下1ケタ 5番
20120612 第621回全国自治宝くじ(ドリーム10(TEN)) 1等 10万円 下3ケタ 484番
                                  :
```

レシピ２　HTMLを動的に作成する。

問題

先程収集した当せん番号のデータに基づき、これをブラウザで照会できるよう、HTML 入力フォームを作りたい。

ただ、抽せん回はドロップダウンメニューで選択できるようにしたい（手で打つのは面倒だ）。そこで、先程の当せん番号データファイルから動的に選択肢（<option> タグ）を生成して欲しい。

解答

例えば Web 掲示板であれば送信されたメッセージに応じた HTML を生成する。このようにして動的に HTML を生成したいことは多々ある。そこで Tukubai にも、このような使い道を想定したコマンド “mojihame” (文字ハメ) が用意されている。

まずテンプレートとなる HTML を作っておく。そこには mojihame が反応するマクロ文字列を埋め込んでおく。このテンプレート HTML と表データを mojihame に掛けると、データがハメ込まれた HTML になるというわけである。

従って、動的に HTML を作りたい CGI スクリプト内にこの mojihame を仕込めばよい。

サンプルプログラム

宝くじの Web フォームを生成するサンプルは、テンプレート HTML(**リスト２**) と CGI スクリプト (**リスト３**)の２ファイルから成る。

リスト２．LOTTERY.HTML : 宝くじ当せんチェック Web フォーム・テンプレート

```
 1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transition
   al//EN" "http://www.w3.org/TR/html4/loose.dtd">
 2 <!-- HTMLというdirectory作成後、その下に配置のこと -->
 3 <html>
 4
 5 <head>
 6
 7 <meta http-equiv="Content-Type" content="text/html; cha
   rset=UTF-8" />
 8 <meta http-equiv="Content-Script-Type" content="text/ja
   vascript" />
 9 <meta http-equiv="Content-Style-Type" content="text/css
   " />
10
11 <title>Open usp Tukubai 宝くじ当せんチェック</title>
12
13 <style type="text/css">
14   * {padding:0px;} /* defaultのpaddingを全て0に */
15   td.komoku {font-size:15px; text-align:right;}
16 </style>
17
18 <script type="text/javascript">
19   function getXMLHttpRequest(){
20     if(!window.ActiveXObject) { //FF,Crome,IE8対応
21       return new XMLHttpRequest();
22     }
23     try{                         //IE7対応
24       return new ActiveXObject("Msxml2.XMLHTTP");
25     }
26     catch(e){                    //IE6対応
27       return new ActiveXObject("Microsoft.XMLHTTP");
28     }
29   }
30   function GetOutput(){
31     var httpReq = getXMLHttpRequest();
32     httpReq.open("POST","LOTTERY.AJAX",true);
33     httpReq.setRequestHeader(
34     'Content-Type','application/x-www-form-urlencoded'
35     );
36     httpReq.onreadystatechange = function(){
37       if(httpReq.readyState != 4   ||
38          httpReq.status     != 200  ) return;
39       document.getElementById('output').innerHTML
40         = httpReq.responseText;
41     }
42     httpReq.send(sendAJAX());
43   }
44   // AJAXの送信情報の作成
45   function sendAJAX(){
46     var data = "NUMBER=" +
47       document.getElementById("number").value + "&";
48     data     = data + "GROUP=" +
49       document.getElementById("group").value + "&";
50     data     = data + "KIND=" +
51       document.getElementById("kind").value;
52     return data;
53   }
54 </script>
55
56 </head>
57
58 <body onload="GetOutput();">
59
60 <form name="INPUT_FORM" id="INPUT_FORM" style="width:80
   0px; height:600px;"/>
61
62 <div>
63
64 <table style="width:800px; height:50px; text-align:cent
   er; border:3px solid #000000;">
65   <tr>
66     <td style="width:100px; border:solid 1px;" />
67       <strong>USP</strong>
68     </td>
69     <td style="font-size:22px">
70       <strong>Tukubai 宝くじ当せんチェック</strong>
71     </td>
72   </tr>
73 </table>
74
75 <br />
76
77 <table style="width:800px; height:120px;" />
78   <tr>
79     <td class="komoku" style="width:360px;">
80       番号６桁
81     </td>
82     <td style="width:400px; text-align:left;">
83       <input type="text" name="number" id="number"
84       style="width:150px;" onkeyup="GetOutput();"
85       value="" />
86     </td>
87   </tr>
88   <tr>
89     <td class="komoku" style="width:360px;">組</td>
90     <td style="width:400px; text-align:left;">
91       <select name="group" id="group"
92       style="width:150px;"
```

Ajax ではお決まりの XMLHttpRequest ラッパー（レシピ３で利用）（19〜29行）

Ajax のリクエスト送受信関数（レシピ３で利用）（30〜43行）

次ページへ続く→

```
 93         onchange="GetOutput();" />
 94         <option value="00" />選択して下さい</option>
 95 <!-- ###GROUP_LIST### -->
 96         <option value="%1" />%2</option>
 97 <!-- ###GROUP_LIST### -->
 98       </select>
 99     </td>
100   </tr>
101   <tr>
102     <td class="komoku" style="width:360px;">
103       宝くじ種類
104     </td>
105     <td style="width:400px; text-align:left;">
106       <select name="kind" id="kind"
107        style="width:350px;" onchange="GetOutput();" />
108         <option value="00" />選択して下さい</option>
109 <!-- ###KIND_LIST### -->
110         <option value="%1" />%2</option>
111 <!-- ###KIND_LIST### -->
112       </select>
113     </td>
114   </tr>
115 </table>
116
117 </div>
```

mojihame コマンド用のマクロ文字列
この区間が表データの行数分複製される

```
118
119 <br />
120
121 <label for="output">当せん可能性:</label><br />
122
123 <div id="output">
124 <!-- ###OUTPUT### -->
125   <textarea cols="97" rows="10">
126 <!-- ###FILEHAME### -->
127   </textarea>
128   <table style="width:800px;">
129     <tr>
130       <td style="font-size:18px; text-align:right;">
131         賞金    ###AMOUNT###
132       </td>
133     </tr>
134   </table>
135 <!-- ###OUTPUT### -->
136 </div>
137
138 </form>
139
140 </body>
141
142 </html>
```

可能性のある当せん番号一覧文字列がこの行に挿入される（リスト４、レシピ３）

可能性のある賞金額の最高値に置換される（リスト４、レシピ３）

リスト４にて、この区間 ("OUTPUT") が抽出され、Ajax のレスポンスとして使われる（レシピ３）

リスト３．LOTTERY.CGI：宝くじ当せんチェッカー Web フォーム・CGI スクリプト

```
 1 #!/bin/sh
 2 #
 3 #######################################################
 4 # LOTTERY.CGI 当せん番号チェックのHTMLフォームを自動生成
 5 # (CGIというdirectory作成後、その下に配置してください)
 6 #
 7 # Written by N.Tounaka(tounaka@usp-lab.com)
 8 #                                  Date : 7 Sep.2012
 9 # Arranged by USP MAGAZINE(mag@uap-lab.com)
10 #                                  Date : 8 Sep.2012
11 #######################################################
12
13 # 環境変数のセット
14 export PATH=$PATH:/home/USERNAME/tukubai/bin
15 export LANG=ja_JP.UTF-8
16
17 # シェル変数のセット
18 homedir=$(pwd); homedir=${homedir%/*}
19 htmd=$homedir/HTML
20 pomd=$homedir/POMPA
21
22 # ヘッダーの出力
23 cat << HTTP_HEADER
24 Content-type: text/html
25
26 HTTP_HEADER
27
28 # ドロップダウンリスト(組番号,抽せん回の2か所)を
29 # はめ込みながら、HTML本体を出力
30 cat $htmd/LOTTERY.HTML                          |
31 mojihame -d_ -lGROUP_LIST - $pomd/GROUP_LIST    |
32 mojihame -d_ -lKIND_LIST  - $pomd/KIND_LIST
33
34 exit 0
```

Tukubai コマンドを /usr/bin や /usr/local/bin 等の標準的な場所に置いていない場合はここで設定

組番号ドロップダウンリストを生成（リスト２の 95~97 行目を置換）

抽せん回ドロップダウンリストを生成（リスト２の 109~111 行目を置換）

解説

HTML テンプレートはそこそこあるが、CGI スクリプトは短く、実質 14 行である。Tukubai を使えばシェルスクリプトでもこんなに簡単に動的 HTML を生成させられる。（生成された Web 画面はレシピ３参照）

ループ文不要、スッキリ書ける

HTML テンプレートの 95 ～ 97 行と 109 ～ 111 行が mojihame 用のマクロ文字列だ。一行に２フィールドあるデータをこの HTML テンプレートと共に mojihame に掛けると各行の n 番目のフィールドが %n にハメ込まれ、それが行数分だけ繰り返される。

結果として、<option> タグ行が自動生成されることとなる。**mojihame のおかげで CGI スクリプト内にはループ文を書かなくて済む**ため、スッキリしたソースコードになる。

Tukubai の名コマンドたち　mojihame と filehame

Tukubai にはレシピ２で出てきた mojihame と、レシピ３で出てくる filehame の２種類の“ハメ”コマンドがある。どちらもテンプレートとなるテキストファイルに、別のテキストデータをハメ込むものなのだが、両者は一体何が違うのだろうか。
先に filehame から説明すると、こちらは単純に文字列をハメ込む。スペースや改行を含んでいてもそのままで、文章をハメ込むのに向いている。一方 **mojihame は、スペース区切りの表データをハメ込む**。だから HTML テーブルの <tr>~</tr> をレコード数に応じて動的に生成できたりする。

filehame：

```
<h1>水瓶座 今日の運勢</h1>
<p>
<!-- FILEHAME -->
</p>
```

```
<h1>水瓶座 今日の運勢</h1>
<p>
今日のあなたは運気最高!!!<br />
特に金運が最高です。<br />
ラッキーアイテムは赤!
</p>
```

mojihame：

```
<table>
<caption>水瓶座 今日の運勢</caption>
<!-- MOJIHAME -->
<tr><td>%1</td><td>%2</td></tr>
<!-- MOJIHAME -->
</table>
```

```
<table>
<caption>水瓶座 今日の運勢</caption>
<tr><td>総合運</td><td>★★★★★</td></tr>
<tr><td>仕事運</td><td>★★★★</td></tr>
<tr><td>金運</td><td>★★★★★</td></tr>
<tr><td>ラッキーカラー</td><td>赤</td></tr>
</table>
```

レシピ 3　AjaxでWebフォームの使い勝手を向上させる。

問題

当せん番号をチェックする時というのは、一桁照合するごとに「このくじにはまだ当せん可能性が残っているか!?」で盛り上がれる。

だから、せっかく Web フォームを作るなら Ajax 等を使って一桁入力するごとにチェックしに行くようにしてもらいたい。

解答

こんなこともあろうかと、先のレシピ２でに掲載した HTML テンプレート（リスト２）は Ajax 仕様で作ってある。

さて、Ajax の場合を含め、サーバー側スクリプトの最初の仕事はブラウザからの CGI 変数を受け取ることだ。これは定番作業であるため、Tukubai にはこれをこなすコマンド "cgi-name" が用意されている。

受け取ったら照合処理を済ませ、結果をブラウザに返す。Ajax の場合は HTML を最初から作り直すのではなく、ブラウザ側で一部を書き換えてもらうための情報を渡すわけだが、Tukubai 流にこなすなら部分的な HTML テキストを作って返す。

サンプルプログラム

レシピ２に掲載したテンプレート HTML(**リスト２**)と右記の CGI スクリプト(**リスト４**)の２つセットでこのレシピのサンプルプログラムになっている。

CGI 変数で渡ってきたくじ番号の部分文字列を元に、その時点で可能性のある「抽せん回＋当せん番号」一覧や、最高金額を返している。

解説

シェルスクリプトや Tukubai による CGI 処理の特徴を２つ述べておきたい。

シェルスクリプトでの CGI 変数取得

CGI 変数は、GET メソッド時は環境変数 "QUERY_STRING" に、POST メソッド時は標準入力に詰められて送られてくる。従ってシェルスクリプトで CGI 変数を取得するには、これらを見ればよいということになる。

リスト４．LOTTERY.AJAX：当せん番号チェックプログラム（Ajax 版）

```
 1 #!/bin/sh
 2 #
 3 ##########################################################################
 4 # LOTTERY.AJAX 当せん番号チェックエンジン+Ajaxインターフェース
 5 # (CGIというディレクトリを作成し、その下に配置してください)
 6 #
 7 # Written by N.Tounaka(tounaka@usp-lab.com) / Date : 7 Sep.2012
 8 # Arrenged by USP MAGAZINE(mag@uap-lab.com) / Date : 8 Sep.2012
 9 ##########################################################################
10
11 # 環境変数のセット
12 export PATH=$PATH:/home/USERNAME/tukubai/bin
13 export LANG=ja_JP.UTF-8
14
15 # シェル変数のセット
16 homedir=$(pwd); homedir=${homedir%/*}
17 htmd=$homedir/HTML
18 pomd=$homedir/POMPA
19 kind_list=$pomd/KIND_LIST
20 lottery_dat=$pomd/LOTTERY_DAT
21 tmp=/tmp/$$
22
23 # ＰＯＳＴメソッドでのデータの受け取り
24 dd bs=$CONTENT_LENGTH  |
25 cgi-name -d_ -i_        > $tmp-name
26
27 # 番号、組、宝くじ種類、名称の取得
28 number=$(nameread NUMBER $tmp-name)
29 group=$(nameread GROUP $tmp-name)
30 kind=$(nameread KIND $tmp-name)
31
32 # HTTPヘッダーの出力
33 cat << HTTP_HEADER
34 Content-type: text/html
35
36 HTTP_HEADER
37
38 # (a)何も選んでなければ何も出力しない
39 if [ $number = "_" -a $group = "00"  -a $kind = "00" ]; then
40
41   sed -n '/OUTPUT/,/OUTPUT/p' $htmd/LOTTERY.HTML |
42   sed 's/###AMOUNT###//g'                        |
43   filehame -lFILEHAME - /dev/null
44
45 # (b)何か選んでいるとき
46 else
47
48   [ $number = "_" ] && number=""
49   [ $group = "00" ] && group=""
50   lottery_name=$(nameread $kind $kind_list)
51
52   #===== 番号チェック ==========================================
53   # 入力先頭番号と当せん下Ｘ桁番号のマッチングを行う。(番号は６桁)
54   #-----
55   # 1:抽せん日 2:宝くじ名 3:等 4:賞金 5:組 6:番号
56   cat $lottery_dat                                                  |
57   # 入力先頭番号と当せん番号を６桁にパディングする                        #
58   awk '{print $0,"XXXXXX"int($6),'$number'"XXXXXX"}'                |
59   awk '{$7=substr($7,length($7)-5,6);$8=substr($8,1,6);print}'      |
60   # 先頭桁から数字をcheck,片方がＸあるいは両方が同じ文字だとOK          #
61   awk '{for(i=1;i<=6;i++){ch1=substr($7,i,1);ch2=substr($8,i,1);
62       if(!(ch1=="X"||ch2=="X"||ch1==ch2)){next}};print}'            |
63   delf 7 8                                                          |
64   #===== 組チェック ===========================================#
65   # 下Xケタ だとＯＫ                                                 #
66   #-----                                                            #
```

Tukubai コマンドを /usr/bin や /usr/local/bin 等の標準的な場所に置いてない場合はここで設定しなければならない
理由は、Web サーバーが動作しているユーザー ID は標準以外のパスが通っていないから

一次ファイルのプレフィップスに自分のプロセス ID を付けることで、不特定多数が同時に実行した場合に起こる競合を回避している

POST メソッドの場合、CGI 変数は標準入力から渡ってくる。そのため dd コマンドで取得する。この時、CONTENT_LENGTH を参照すると何バイト読み取ればよいかわかる

cgi-name コマンドを通したこのファイルには、ブラウザから渡ってきた CGI 変数が URL エンコードを解かれ、変数名と値の２列から成るテキスト表が格納される

nameread コマンドを使うと、目的の CGI 変数が簡単に取り出せる

次ページへ続く→

例えばTukubaiでPOSTメソッドから受け取るなら、リスト4の24行目のように、**ddコマンドから環境変数CONTENT_LENGTHの分だけデータを受け取り、cgi-nameコマンドに流す**。これで、URLデコードと受け取った変数を名前と値から成るテキスト表ができるので、これを適宜処理すればよい。

Tukubai流のAjaxデータ生成

Ajaxというと、その語源からブラウザにはXMLで結果を返すことが多い。だがTukubai流にやるならそうではなくて部分的なHTMLテキストを作り、これを返す。これならブラウザは、innerHTMLメソッドを使って返された文字列を流し込むだけで済む。**なるべくシェルスクリプトを使い、JavaScriptのコード量は抑える**という方針である。

リスト4の90行目〜を見てもらいたい。元のテンプレートHTMLから部分的に抜き出し、そこに結果をハメ込んでいる。こうして、部分的なHTMLテキストを生成している。

```
67   # 1.XX組 / 2.組下1ケタX組 / 3.各組共通 / 4.下Xケタ            #
68   LANG=C awk '{if($5~/^下[0-9]ケタ$/){print; next}              ¥
69   # 入力組と当せん組を2桁にパディングする                         #
70       else if($5~/各組共通/){str="XX"}                          ¥
71       else if($5~/組下1ケタ/){str="X"substr($5,14,1)}           ¥
72       else if($5~/[0-9][0-9]組/){str=substr($5,1,2)};          ¥
73       grp='$group'"XX";                                        ¥
74   # 先頭桁から数字をcheck,片方がXあるいは両方が同じ文字だとOK #
75       for(i=1;i<=2;i++){ch1=substr(str,i,1);ch2=substr(grp,i,1);
76       if(!(ch1=="X"||ch2=="X"||ch1==ch2)){next}};print}'       |
77   #===== 宝くじ名チェック ====================================#
78   awk '$2~/'$lottery_name'/{print}'                > $tmp-result
79
80   # ===== 当せん最高金額レコードを求める ======================
81   grep -E "[0-9.]*億円" $tmp-result |
82   sort -nr -k4,4                    > $tmp-work
83   grep -E "[0-9]*万円"  $tmp-result |
84   sort -nr -k4,4                   >> $tmp-work
85   grep -E "[0-9]*円"    $tmp-result |
86   grep -vE "*万円"                  |
87   sort -nr -k4,4                   >> $tmp-work
88
89   #===== ファイルハメ ========================================
90   sed -n '/OUTPUT/,/OUTPUT/p' $htmd/LOTTERY.HTML                 |
91   sed 's/###AMOUNT###/'$(head -1 $tmp-work | self 4)'/g'        |
92   filehame -lFILEHAME - $tmp-result
93
94 fi
95
96 rm -f $tmp-*
97 exit 0
```

リスト3の"OUTPUT"で囲まれた区間（124~135行目）のみを取り出す

リスト3の131行目にある###AMOUNT###を最大賞金額に置換する

リスト3の126行目（"FILEHAME"のある行）を可能性のある当せん番号一覧に置換する

おまけ　宝くじ当せんチェックアプリを使ってみる。（デプロイメント）

せっかく作ったのだから実際に動かしてみよう。まず設置手順を記す。

設置（デプロイ）手順

1．Tukubaiインストール

これ無しには始まらない。root権限の与えられていないレンタルサーバーにインストールする場合は、make install時にLOCALBASE変数の設定が必要になるだろう。手順は3ページを参照のこと。

2．リスト1～4の配置

下記の位置関係でファイルを置く。HTMLファイル以外は全て実行権限を付けておく。

尚、ファイルは全部大文字にする。これはユニケージ開発手法における作法である。

```
./┬CGI/┬LOTTERY.CGI      （実行権限付加）
  │     └LOTTERAY.AJAX   （実行権限付加）
  ├HTML/─LOTTERRY.HTML
  ├POMPA/
  └SHELL/─LOTTERY.DB     （実行権限付加）
```

3．リスト3，4にパスを設定

両リストの冒頭に環境変数PATHを設定している箇所がある。Tukubaiをrootで入れていない場合は、ここにTukubaiコマンドへのパスを記述しなければならない。

4．cron設定

LOTTERY.DBをcron等で定期実行させるようにする。或いは定期的に手動実行する。

5．Webサーバーの設定

手順2で置いたCGIとAjaxファイルが実行できるよう、設定ファイル（.htaccessやhttpd.conf等）を設定する。

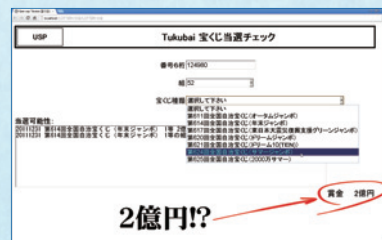
お待ちかねの「あそびかた」

さあ、ジャンボ宝くじを買ってきて抽せん日が過ぎたらこのソフトでチェックしよう！

ブラウザでLOTTERY.CGIを開いたらくじ番号→組番号と、まぁあせらず入力していって欲しい。すると、可能性のある当せん番号と貰える可能性のある最高賞金額が画面下に表示されていく。

打ち込みを進めるたび、絞り込まれ、だんだんと賞金額は減っていくことだろう。でも、運よく2億円のまま組番号まで入れ終えたとしようじゃないか。そうしたら最後に「ところでそのくじ、いつ抽せんされたくじ？」という質問に答えるべく、抽せん回を選ぼう。

その結果、表示が消えたアナタ！おめでとう、ハズレだ！でも**「ちくしょー！この回にこのくじを買っていれば億万長者だったのにー！」という悔しさを自慢できる**ぞ。

これが、このアプリの正しい使い方なのだ。

やった、2億円じゃん!! ……ん?
俺が買ったのはサマージャンボなんだが……、
こ……これは、もしかしてドボンなんじゃ

笑顔が素敵な恭子さん。
可愛らしい上、とても勉強熱心な方だと聞いて、取材に伺いました。
それもそのはず、西本さんのお仕事は Linux ～ AIX のサポート。
生半可な知識量では毎日の業務をこなせません。
話を聞いていると、彼女の瞳に惹きこまれます。
ひとつひとつ丁寧に仕事をしているのだろうな、と思わせるような真剣な眼差しです。その反面、ふと彼女の手を見てみると素敵なネイルアートが。
バリバリ障害対応している傍ら、女の子らしい色使いのファッションと愛嬌のある笑顔で職場も同時に癒しているのでしょう。

Profile　西本恭子さん
TwitterID：kyoko_0310

Q）Linux との出会いを教えてください。
A）会社に入って配属された部署が、Linux（RedHat）のサポートの部隊でした。
　プログラミングですが、高専で C 言語や VB などの基礎を勉強しました。
　しかし、学業でプログラミングを学んでいたときは、自分が触っていたパソコンに入っている OS が Linux だとか意識した記憶がないくらい、初心者だったと思います。
　どの OS の環境でやっているかもわかっていなかったんです。今考えると、一方的に学ぶような場所では、技術に興味が沸かなかったんですよね。「勉強したい」と思い始めたのは仕事で問題を解決しなければならない状況になってからです。
Q）実践でたくさん学ばれたのですね。Linux の好きなところは？
A）オープンソースなところ。（笑）商用 Unix と比べると、ネットにある情報量が圧倒的に多いんです。
　インターネットで調査できるのは大きな魅力です。あと、自分でソースコードを見て調べられる。CUI なところも好きです。
　トラブルがあったとき、Windows では自分で解決するのが難しくて・・・。Linux は公開されているのでサポートしやすい！
Windows のトラブルシューティングは業務外なので良かったです。ネットで解決情報、なかなか探せないですからね。
Q）マイクロソフト社にはまだお世話になっていない西本さん、これから Linux とどうやって付き合っていきますか？
A）仕事や勉強の基礎となっているので、このまま続けると思います。
　入社して７年目の今は、IBM の商用 Unix である「AIX」のサポートのお仕事をしています。AIX で問題が起きた時に Linux でも起きるか調べますね。根本的なところは一緒なので、Linux が解決の糸口となることがあります。そう思うと、Linux を社会人になってから学んだ経験が私の基礎になっていますね。
　サポート業務のエンジニアは AIX と Linux を出来るように、会社が検定試験受験などを色々と支援してくれます。勉強しやすい環境で会社に感謝しています。
Q）勉強を応援してくれる会社は心強いですね。先輩と後輩の関係はどうやって保っていますか？
A）悩まずコミュニケーションしています、おしゃべりして問題解決。（笑）
　自分もよくしゃべるほうですが、先輩もよくしゃべるので、よく雑談していますよ。
上下関係をあまり気にせず、積極的にご飯食べにいったりしていますね。とくに親会社の先輩にはサポート対応のエスカレーション先（※）ということもあっていろいろ教えてもらえるので、よく飲みにいったりしています。頻繁にコミュニケーション取ることで、いろいろと助けてもらうんです。（満面の笑顔、そんな印象でした）

※ 簡単に解決できない問題を、一次受けから二次受けへというように、エスカレータ形式で問題分析・解決部署をレベルアップさせること。

きらりん☆

高専出身の恭子さん。もともと理系で容姿に恵まれ多彩だけれど勉強に対する努力は忘れない。
そんな彼女のつぶやきは「先輩に助けられた」「〇—〇〇がやさしい」とか、感謝の言葉が多いような気がします。ハートが良い上、こんな素敵な笑顔がかわいらしいのだから、周りの人達は自然と彼女に手を差し伸べるのだろうな、とちょっと年増なおねえさん編集担当は思うわけです。よしよし。

本コーナーでは自薦・他薦を問わず（そして、Linux に限らず？）Tech 好きな女子を大募集しています。スーパーエンジニアだけではなく、これから Linux を勉強しようとしている向学心のある女性でも大歓迎です。本誌を通してエンジニア友達作り、恭子さんのように交流の幅を広げてみませんか？　詳しくは mag@usp-lab.com　までご連絡ください☆

LL DECADE イベントレポート

2012年8月4日 @銀座ブロッサム

Perl、PHP、Python、Ruby、JavaScript――。
テキスト処理や Web アプリケーション開発などに不可欠な、コンパイル要らずの軽量言語、Lightweight Language。
近年は、業務システム領域において C や Java に劣らぬ存在感を示すほか、またモバイル系のゲーム、家電などの組み込みシステムから、かつてスパコンで行われていたようなシミュレーション、科学技術計算領域まで、その用途に拡がりを見せています。LL を支持するプログラマもこの 10 年で増えました。
こうした LL の盛り上げに一役買っているのが、今年で開催 10 年目を迎える Lightweight Language イベント。
ちなみに、LL の源流を遡るとシェルにぶつかる、ということで、本誌もこのイベントを応援しています。2012 年 8 月 4 日、銀座ブロッサムホール（東京）において開催された第 10 回記念の LL イベントでは、LL 魂を持つ各言語界の雄、総勢 20 名以上が登壇し、熱のこもったトークを繰り広げました。一日の様子を振り返ります。

基調講演

言語を取り巻くエコシステムに興味がある。互いに刺激し、足りないところを補っている関係が面白い

宮川達彦さん

乱立する Web アプリケーションフレームワークと Web サーバーをつなぐインターフェース PSGI の仕様作成、それを実装した Plack の開発で知られる宮川さん。Perl を中心に多くの言語にアンテナを張り続けていますが、この数年、仕事や趣味で主に Python、最近は Ruby を使う機会が多いそうです。「Ruby はオブジェクト指向が付け足しではなく、綺麗にまとめられている。手錠をはめられていない感じでプログラミングできるのがいい」と宮川さん。「ただし、僕はプログラミング言語そのものが好きというよりも、各言語が織りなすエコシステムのほうに興味があります。相補的に発展する関係性が面白い」。

Perl
Descriptive, boring, ::Simple
Python
Descriptive, confusing, py*, *.py
Ruby
Fancy, creative, chaotic

名前のつけ方に個性が出る

PSGI や Plack の着想も、Python や Ruby からインスパイアされたもの。その Plack がさらに、言語の種類や海を越えて、様々なプログラマにフォークされています。「例えば、Rails4.0 の開発をしている人たちを見ると、かつて僕らが歩いた道を同じように辿っている既視感を覚えます」。

"You don't need someone's permission to write code" と宮川さん

CPAN のようなリポジトリは誰か一人で完結することはありえません。「これ、スゲーな」と思うコードを互いにどんどんフォークし、その過程で言語が互いに洗練されていく。それこそがオープンソースの醍醐味、と語る宮川さん。「その意味で、GitHub はとてもアナーキー。プログラミングを民主化した」とそのインパクトの大きさを指摘します。

「言語のコミュニティはいろんな仲間が参加する鍋パーティと似ています。あまり早く着いてもすぐ食べられない。手伝わされることもある。ところが少し遅れていくと、うまい具合に食べ頃になっている（笑）。早過ぎるよりちょっと遅れて参加するくらいでちょうどいい。ただし、他の言語のコードをフォークするときは、" コピペのミス " に注意」と自らの失敗談を語り、会場の笑いを誘う一幕も。

ところで、いろいろな言語に関わることで視野が広がったという宮川さん。

「カンファレンスに参加すると、肩書きなどではなく、『コアのあのコードを書いたミヤガワ』という感じで声をかけてくれる。プログラマにとってコードは、自分自身。リポジトリはその時々に自分が考えていたことや興味を持っていたことを示すタイムラインのようなもの。転職した際も、CPAN の URL が履歴書代わりになった」と打ち明けます。

来場者に「LL イベントのような、異なる言語間のプログラマの交流には積極的に参加されることをおすすめ

My code
= What I really am

プログラマにとってコードは名刺代わり

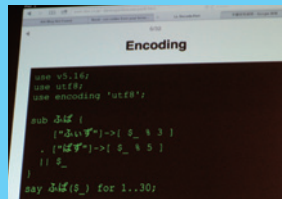
します」と呼びかけた宮川さん。いま注目していることは何か、という講演後の公開質問に「僕は面白い仕事をしたい。LL はその可能性を拡げてくれます」と答えました。

Language Update Decade

Perl、PHP、Python、そして Ruby という 4 つの LL における近年の状況を、各言語界を代表する強者が語るセッション。今回の顔触れと席順は 2003 年に開催された LL Saturday の再現です。司会も当時と同じ法林さん。4 大言語のエキスパートが語る、この 10 年間の変化、そして未来の姿とは。

Perl

小飼弾さん

Perl はこの十年で、新機能がいろいろと追加されています。しかし、これまで書いたコードを動かせるように気を遣っているのが Perl のよいところ。当たり前に聞こえるかもしれないですが、他の LL では必ずしも、過去のコードがそのまま動くとは限りません。互換性に関する Perl のコンセプトは 10 年後もおそらく引き継がれているでしょう。Perl プログラマは他の言語を扱うのと違い、これから先、どれだけ仕様が変わるだろうとむやみに心配せずとも、プログラムの開発に専念することができるはずです。

機能が拡充する Perl5 を中心に解説

ところで、LL 全体を見渡せば、「サーバーサイドでも動く JavaScript と、その他大勢」という観があります。しかし、JS には CPAN に相当するリポジトリがない。お世辞にもシェルフレンドリーとは言い難い。Perl はやはりよい選択肢ではないかと思います。

PHP

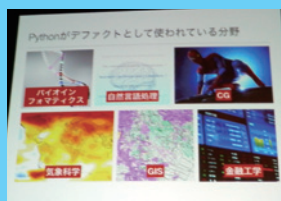
廣川類さん

PHP がブレークしたのは、バージョン 4 からです。2006 年頃でも半数以上のユーザーが、PHP4.x を使い続けていました。2004 年 7 月に発表されたバージョン 5 からは、SOAP、Web サービス、DB の強化など、オブジェクト指向の深化が図られました。2005 年に PHP が誕生から 10 周年を迎え、それを機に、東京だけでなく、関西、北海道でも PHP カンファレンスが開催され、賑わいを見せました。この 10 年を振り返ると色々ありましたが、PHP 作者であるラスマス自身によるバージョン 6 のキャンセル、という出来事が心に残っています。言語の開発というのは一筋縄ではいかないなと思いました。ところで、プログラミング言語には色々ありますが、大事なのは、きちんと使いこなせることです。PHP 作者自身が、PHP は歯ブラシのようなもの、と評しています。毎日、当たり前に動いて、身近で役立つ自然な言語、それが PHP です。

受け継がれる PHP 作者のコンセプト

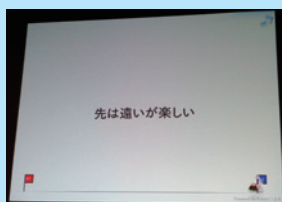
Python

柴田淳さん

今から 10 年前、日本では Python を使っている人が少なく、プログラマの間では珍獣扱いでした。ただ欧米では注目度が高く、Google で使われていることも話題になりました。とはいえ、この 10 年、Python 自体はあまり変わっていません。2003 年にリリースされたバージョンにはすでに今見ても遜色ない機能が実装されています。しかし、利用者の層は著しく拡大しました。Linux のインストーラー、デスクトップは Python で作られています。DropBox や YouTube もそうです。意外なことに、ノンプログラマにも Python が利用されています。安価な PC サーバーを使って並列計算をこなす科学技術計算用途もそのひとつ。日本の気象庁でも使われています。Python が他の LL と一線を画す点は、ライブラリが充実していること、そして言語の仕組みや使い方がシンプルであること。Python プログラマはこの先 10 年も食べていけるのではないかと思っています。

用途が拡がる Python

Ruby

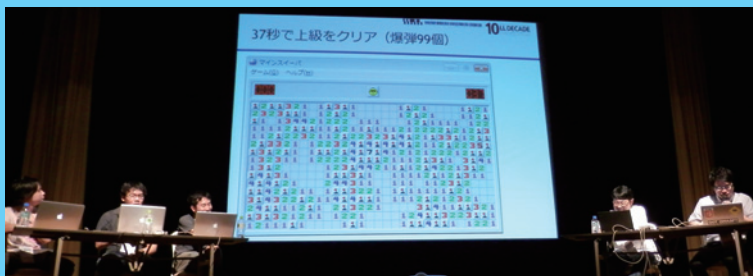
まつもとゆきひろさん

2003 年の LL Saturday に出させてもらった時、Ruby のバージョンは 1.8 でした。2007 年に、コアを大幅に置き換えた Ruby1.9 を出しましたが、いまだに多くのプログラマが、1.8 を使っています。当時は、Perl や PHP など「P」で始まる言語が優勢で、Ruby は肩身が狭かった。流れが変わったのは、2004 年に発表された Ruby on Rails からです。当時、Java と Ruby のどちらの生産性が高いか、といった議論もありました。それから 10 年近くが経ち、フルタイムで給料をもらえる Ruby プログラマが増えました。なぜ、Rails や Ruby が広まったか。うまく言葉にしがたいのですが、使っていて気持ちがいいからだと思います。一般に LL は道具もしくは黒子で表には出てきません。しかし、Ruby ユーザーは他の LL ユーザーにはないほど言語に対する愛情を示します。色々なコミュニティのシンボルになっている。まもなく Ruby も誕生から 20 年です。それを機に、Ruby2.0 を出そうかなと思っています。

まもなく誕生から 20 年

プログラミング言語処理系を自作して分かったこと

next Perl5 プログラミング言語 Tora 開発者の Tokuhiro Matsuno さん、「Perl6 の実装にはまだコレというものがない。処理も遅い。とはいえ Perl5 の使い勝手はシンプルではない。自分なりに理想を追求して、Perl5 から 0.5 くらいアップした言語を作った。こうした取り組みを通じて Perl5 や Perl6 の開発に貢献できれば」と発言。一方、@Constellation さんは、ECMA Script の次期バージョン（ES.next）について言及。「ES.next エンジンは仕様に忠実な実装と、処理の高速化を目指しています。ECMA Script の未来は明るい！」と語り、JSX を開発した @kazuho さんは、「モバイル向けのゲームの開発にあたって高速化、デバッグ品質の向上、ソースコードの見やすさが求められている。JSX ではより速く安全で簡単な JavaScript を目標に掲げ、今後はライブラリの拡充を図りたい」と展望を述べました。

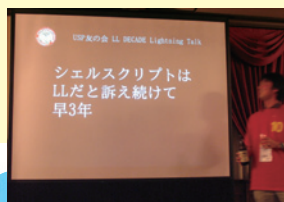
組み込み向けの mRuby を開発中のまつもとゆきひろさんは、「cRuby ではできない領域に拡げていきたい。太陽光発電の管理システムや、消費電力の見える VPN ルーターなどのアイデアや実装例も少しずつ現れている。正式には 2013 年の発表をご期待ください」と意気込みを披露しました。

Perl6 の仕様と動向について解説を行った司会の竹迫良範さんには来場者から「自身で言語を開発しないの？」という質問が飛ぶワンシーンもありました。

懇親会レポート

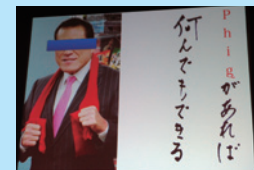
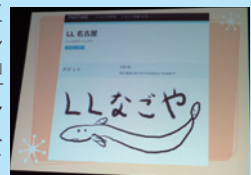
LL イベント本編の勢いそのままに始まった懇親会。ここではスポンサー企業などによる Lightning Talks が行われました！

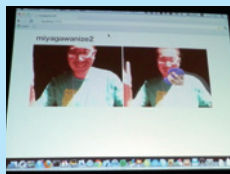
どろくさい言語と LL のイイトコ取りをしたという東北大学電子通信研究所が、開発中の SML# について講演。

LT のトリを務めたのは、USP 友の会。Tukubai ハンズオン、bash 勉強会、IT パパママ部、ペアプロ合コンなどの企画で積極的に来場者を勧誘。

PHP カンファレンスは、9 月 15 日に大田区産業プラザで開催。9 月 14~16 日は、札幌 Ruby 会議。9 月は LL 絡みのイベントが目白押し。

俺たちの継続的 hogehoge は始まったばかりだ！

アプリケーションの品質改善や納期短縮に向けて各社が取り組む継続的インテグレーション（CI）の現状とは。Jenkins をはじめとする各種 CI ツールを活用したさまざまなテスト事例が披露されました。

「出版社のシステムでビルドされた原稿をテストする際に Jenkins を活用」cactusman さん（BeProud.inc、日本 Jenkins ユーザ会）、「LL はダックタイピングを活用することで、いろいろなモックを作れる」太田健一郎さん（TDD 研究会、Android テスト部）、「JsTestDriver を使い、各種実ブラウザにおける JS テストを効率化している」@kyo_ago さん（KAYAC）、「社内を巻き込む工夫を行ったところ、テストコードを書かないメンバーが激減した」高倉利明 @TrinityT さん（株式会社アピリッツ）、「テストに限らず、日々のタスクをスクリプトで自動化するのが重要」きょんさん、「許可を取らずに始め、成果を出してから上司に見せる、そして楽しむことが CI 浸透のコツ」山城裕 @yamashiro さん。

皆さん、いろんな工夫を重ねているんですね！

Lightning Talks　～ 5 分間一本勝負～

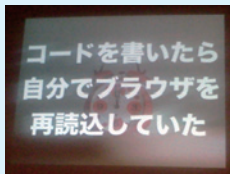
ゆーすけベーさん

人物写真と、宮川達彦さんの手にする「紫色の何か」を合成して表示する Perl プログラム「miyagawanaize」そのバージョン 2 が公開。0.5 秒おきに写真を読み込み、アニメーション化。その奇妙な動きに会場は大爆笑。

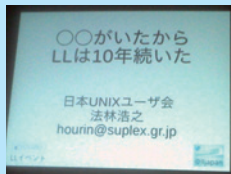
角谷信太郎さん

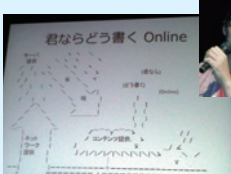
日本 Ruby の会で理事を務める角谷さん。「僕は Geek にはなれない。でも、そうした人を集めてイベントを 10 年やってきました。チャレンジし続けるには、いつも一歩上のレベルに噛みつくぐらいがちょうどいいと思います」と話しました。

川崎有亮さん

シリコンバレーではプログラマの人件費が高騰し、クラウドソーシングが流行。この時代を生き抜くために必要なのは、流暢な英語？　基盤的なソフトを作る技術？。いや「紫色の何か」ということで、画面の青系色を緩和する目に優しいプログラムを公開。

庄司嘉織さん

ブラウザのリロードはムダ、ということで処理を自動化するスクリプトを CoffeeScript で自作した庄司さん。最後は、山城裕 @yamashiro さんの彼女募集。そっちが本題か？

法林浩之さん

スピーカーとして初めて参戦の法林さん。本家米国の LL は 2 年で終了。なぜ、日本では LL が 10 年続いたのか。イベント名を工夫し、リングの上でやったり、トーナメントをやったり。これからの 10 年にも大いに期待しています。

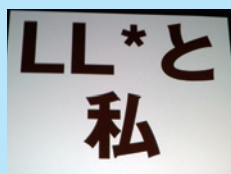
LLNOC チーム

空気や水のように使える、無線 LAN 環境を会場内に構築してくれたネットワーク部隊。今回も約 300 人の来場者が利用したそうです。これまでのイベントで蓄積したノウハウやエピソードを報告。

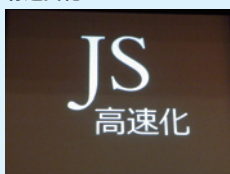
yuroyoro さん

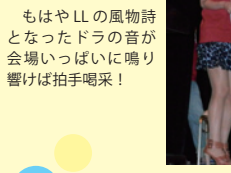
株式会社スケールアウトで広告配信を手がけている yuroyoro さんは、音声にフォーカスした話。最後は、自作の音声認識ツールに「バルス！」と呼びかけ、PC をシャットダウン。見事時間内に講演を終了。

ただただしさん

Ruby でブログツール tDiary を開発してきたただささん。「Ruby のバージョンを意識することなく使える。初期のユーザーを置いてけぼりにせず、新機能を追加してきた。25 年使えるものにしたいから後 14 年は続ける」と宣言。

高橋征義さん

LL には、初期の頃からスタッフとして関わり、その足取りを見てきた高橋さん。「学校のような場所。イベント慣れしていなくても足を運びやすい。プログラマならば言語を問わず大歓迎」。この日は、自身が会長を務める日本 Ruby の会についてもアピール。

竹迫良範さん

最後の LT を飾ったのは竹迫さん。サーバーサイド JS が注目される中で、JIT コンパイラの脆弱性を指摘。JIT-Spray 攻撃を実演すると、悪意あるコードがおなじみの爆弾アプリ（無料）を起動し、会場は大爆笑。

司会を進める郷田まり子さんと、ドラ娘のショウジユウコさん

もはや LL の風物詩となったドラの音が会場いっぱいに鳴り響けば拍手喝采！

ロビーで展示した書籍見本やグッズがあたる抽選会のボール投げ

LL イベントの盛り上がりは、業界における LL の浸透と軌を一にします。

まつもとゆきひろさんが指摘されていました。「どの言語の技術がいいとか悪いという話ではなく、言語を使って何かを変えていこうと、人々をドライブする熱意こそが、世界を動かすのではないか」。

大切なのは、言語の壁を越えた、参加者同士の交流。互いの言語をリスペクトし、刺激し合うこと。LL の世界はグローバルです。LL イベント、次の 10 年が始まります。

特集２

デール・ダハティ氏に訊く、技術者哲学

私たちは単なる
消費者（Consumer）ではない。
誰もが世界を変える力を持った
創造者（Maker）だ。

「Hacks」や「クックブック」シリーズで知られるオライリー・メディア（O'Reilly Media, Inc）創設者の一人、デール・ダハティ氏が、2012 年 6 月、来日した。
目的は日本の Maker が繰り広げるムーブメントの視察と、Maker 達との交流だ。
「Maker（メイカー）」とは、デール氏曰く、「文化の創造者」。道具を駆使し、世界を探索（ハッキング）する人。単なる消費者ではなく、作る喜びを知っている人。世の中を楽しませ、社会を変える人。
「皆さん一人ひとりが Maker です」とデール氏は語る。その言葉に込めたデール氏の思い。そして、ソフトウェアエンジニアを含めたこれからの時代を担う Maker に必要なものとは何か。
（聞き手：松浦智之、まとめ：柏崎吉一）

――プログラミングやシステム運用管理などで UNIX を学ぶ人は誰しも一度、オライリー・メディアの本を手にしたことがあると思います。その一つ、「sed&awk プログラミング」の著者であるデールさんが、ティム・オライリー氏とともに発行人を務める、オライリー・メディアを立ち上げた経緯や書籍執筆の動機を教えていただけますか。

　もともと、私とティムは、文学少年でした。いつか自分たちの手で出版社を作りたいという夢を抱いていたのです。

　テクニカルライターとして仕事をこなすようになっていた 80 年代、私たちの業界にもコンピュータ革命の波が押し寄せていました。私たちが UNIX に注目したのは、これを使えば出版社を作れると考えたからです。UNIX 上で、書籍や雑誌における文字組みや図版のレイアウトデザインを行い、誌面の入稿データを作る。後は、レーザープリンターなどで印刷し、製本すれば自分たちの手で本や雑誌を作れると考えました。

――DTP の草分けだったのですね。

そうです。ただし、私たちはエンジニアではなく、テクニカルライターなのでコンピュータの知識はほとんど持ち合わせていませんでした。しかし、勉強すれば何とかなる、自分たちもきっと使いこなせると楽観的に考えていました。そして勉強の過程で身に付けたことを逐一記録しておき、それをもとにテーマを絞った UNIX ユーザー向けのハンドブックを作成しよう。最初はそんなノリだったのです。

「なければ作る」。これはオライリーの創業時からの企業精神です。取扱説明書やシェルのマニュアルと首っ引きになって、プログラムをいくつも作りました。そんな風にして原稿をまとめ、刷り上がった本一冊の値段はたしか 5 ドルでした。ニューヨークで開催された UNIX EXPO でそれを完売し、手にしたお金で、ティムと私は、ブロードウェイで観劇した思い出があります。

アマチュアたちの創造性に強く共感

――「作る」ということに興味を持ったのはその時からですか。

はっきりいつ、とは言えませんが、私は次第に、本の制作に限らず、『" 何かを作る " という行為の楽しさ・充足感は、特別な人達だけが独り占めするような感情であってはならない』と思うようになりました。作り手は特権的な存在ではない。作る喜びは誰しも味わえる。私たち一人ひとりが Maker であると思うに至ったのです。

「ある人が Maker かどうか」を見分ける簡単な方法があります。その人の目の輝きです。何かの製作に没頭している時、人はたいてい生き生きしています。ではなぜ、一心不乱に作業しているのか。回りの人に見せたい、笑わせてみたい。心の中から湧き上がってくる、抗しがたい衝動のようなものがモチベーションだと思います。私たちがオライリー・メディアを立ち上げた時もそんな感じでした。

そういった Maker たちを数多く取材しているうちに、オライリー・メディアは「Make」という雑誌を生み出しました（P.40 囲み記事）。「Make」で取り上げているのは、世の中をハッキングする人々です。Maker と Hacker はほとんど同じ意味だと私は思います。Maker のほうがその指し示す範囲が Hacker より、広いだけです。

――「Make」初の日本版（2006 年 8 月発行）では、「ビデオデッキを改造して猫の給餌機を作る」という記事があり、実際に動く作品のメカニズムなどを紹介していますね。

取り上げる作品には実用的なものもありますが、一見しただけで、何のために使うものかよく分からないものも多々あります。とはいえ世相を見事に反映し、風刺が利いているものが少なくありません。儲けや名誉のためではなく、内から込み上げてくる創造したいという欲求に突き動かされてできあがった作品がほとんどだと思います。

最近、私が興味を持った作品は、米国人が自作したガイガーカウンターです。昨年の福島第一原子力発電所の事故をきっかけに、自分の身の回りにあるものを組み合わせて自作したと聞いています。私がこの作品に興味を持った理由は、Maker 同士が築いた、海を超えたネットワークにあります。日本の出来事が契機となって米国の Maker を動かす。自ら考える個人がネットワークでつながり、作り方などの情報をシェアし、コラボレーションする。新しいものづくりの形態です。ガイガーカウンターはあくまで一例ですが、Web 上に発表されたある作品がボーダレスに次々と他の Maker を刺激し、Maker の一大ムーブメントとして世界各地に波及していくことがあります。

そうした Maker 達が年に一度、一堂に会する場があります。Maker Faire です（P.39 囲み記事）。

――どんなイベントなのでしょうか。

Maker 達のお祭りです。作品を会場一杯に展示します。

ちなみに米国には、Renaissance Faire という、ヨーロッパの中世時代の人々の格好をして楽しむコスプレ・イベントがあるのですが、Maker Faire もそれと同じで、まったく堅苦しいイベントではありません。子供も大人も楽しめます。ぜひ一度足を運んでみてください。人生観が変わりますよ。

今年 5 月にベイエリアで開催された 7 回目の Maker Faire では、2 日間でのべ 11 万

Maker Faire

～ Maker 達の作品が一堂に会するお祭り～

今年で 7 年目を迎える Maker Faire は、これまでサンフランシスコ・ベイエリアを中心に、デトロイト、ニューヨークなどでも開催されてきた。これと並行して日本では 2012 年 11 月に、第 8 回目となる「Maker Faire Tokyo」が開催される予定（第 1 ～ 7 回までは「Maker：Tokyo Meeting」という呼称だった）。米国で始まった Maker Faire のムーブメントは、日本を含む世界 60 カ所に飛び火している。

（写真は Maker Faire BayArea2012 のハイライトから抜粋 http://makerfaire.com/bayarea/2012/index.html）

Make
～ジャンル不問の DIY 雑誌～
英語版は 2005 年から通巻 31 号が発売。それを翻訳した日本版は 2006 年 8 月の創刊から 2011 年 4 月まで通算 11 号が発売されている。

専門業者に頼らずとも、個人の力でいつの間にかこんなものまで作れる時代になったのか、ということを見せつけている雑誌です。本物の宇宙へ送る人工衛星まで作ってやがります。国家が莫大な予算を投じて作ってるのを尻目にですよ！今や必要なのは金じゃない。才能を無駄遣いして神と呼ばれる存在になってやるという情熱くらいなものですよ。（松浦）

人が集まりました。すごい熱気でした。

「この世界は、僕たちとさほど能力的に変わらない人たちの手で作られている」。こう言ったのは、スティーブ・ジョブズです。コンピュータは世界を変える、彼はそう確信して、スティーブ・ウォズニアックと同様に大学を中退し、マニアが集まるカリフォルニアのホームブリュー・コンピュータ・クラブに顔を出すようになります。そこでコンピュータ開発に残りの一生を捧げると誓った。そのクラブはきっと Maker Faire のようなものだったと思います。

彼らが最初につくった Apple Ⅰ。その筐体は木製でした。よく見ると、欠けているキーがあります。完璧ではない。しかし作り手の情熱がひしひしと伝わる一品です。Maker Faire の会場では、この “Apple Ⅰ” を彷彿させる作品をよく目にします。外観はエレガントではないけれども、内に秘めた可能性を感じさせる。Maker Faire に出展されている今日の作品が少しずつ改良され、30 年後には “Apple Ⅰ” から “iPad” に至るような大変身を遂げているかもしれません。

そういう作品を見ながら、「一体どこからこういうアイデアが湧いてくるのか」と、作り手の思いを想像するのはとても楽しいものです。

日本人も参加しています。北村満さんが展示された自作のプラネタリウムは感動しました。私がドームの中に入ると、映画「スターウォーズ」の何シーンかを影絵で即興表現してくれたのです。Maker Faire は、アートと技術の橋渡しをするクロスカルチャーの場だと思いました。

ツールセットより大事なのはマインドセット

――7 年前の Maker Fair と比べて、何か大きく変化した点はありますか。

アイデアを形にするための「道具」がいまほど入手しやすい時代は過去になかったでしょうね。PC にしてもそうです。立体加工を行うためのレーザーカッター、小型の 3D プリンターしかり。

先日、ジョージア工科大学の技術系研究室を訪ねましたが、やはり学生たちがこれらのツールをごく当たり前に使って研究や製作活動を行っていました。かつては専門家しか、あるいはお金持ちにしか使えなかった技術が、ほぼ誰でも使える。技術の民主化が起きているのです。道具へのアクセス権については、プロとアマの差がほとんどありません。学校や家の中が小さな工場になりえます。Maker space（P.41 囲み記事）というのは、そうした道具や材料を一式揃えて、子供たちが遊べるようにした工房です。

でも、仮に道具が手に入らない場合でも、悲観する必要はありません。ないものは工夫して編み出せばいいのです。それがコストの削減につながることもあります。こういう発想の仕方こそが、Maker 精神です。大事なのは、ツールセットではなく、マインドセット。ツールでもなんでも作ろうという情熱と遊び心があれば、たいていの困難を乗り越えられます。

DIY（Do It Yourself）の意味するのは、決して孤独、独りぼっち、ではありません。最初はたとえ一人で作ったとしても、成果を他の人に見てもらう。するとアイデアを付加してくれたり、改良してくれたりする仲間がだんだん増えていきます。

オープンソースソフトウェアは、そのようにしてコミュニティの中で日々進化していますよね。同じように、3D プリンターや卓上工作機械といったハードウェアも日々改良が重ねられているのです。

改良を重ねて世の中が変わる

――デールさんは、2011 年にホワイトハウスから「Champion of Change（変革の旗手）」という表彰を受けましたね。

実のところ私は、官僚や教育関係者などのおエライ人々から褒められることにあまり関心がないのです。そうした多くの大人は、内的衝動に駆られて世の中を探索する Maker ではなく、消費者あるいは何かの目的のために道具を使う利用者に留まっていることが多いからです。

時々、大学などに招かれて講義することがありますが、そこで私にとって本当に興味あるのは、学生達の目の輝きです。学生と話を

デール・ダハティ
（Dale Dougherty）
●オライリー・メディア創業者の一人。「sed&awk」の執筆者。雑誌「Make」の編集者・発行人。現在、オライリー・メディアの Maker Media 部門を率いる統括マネージャーである。1993 年に、世界初となる商用 Web サイトを立ち上げた人物としても知られる（同サイトは 1995 年に AOL に売却）。

していると、思いもよらない奇抜なアイデアが飛び出してきます。失敗を恐れずにチャレンジする姿勢を見るのが私は大好きなのです。

そもそもこの社会全体が、発明や改良の連続によって作られています。最初からこのように出来上がっているわけではありません。また、誰か一人の手によって作られたわけでもない。誰かが作ったものを後の世代がちょっとずつ改良していくだけです。完成は永遠にありえません。

物心ついたばかりの子供の目に飛び込んでくる世界は、あたかも "すでに出来上がった完成品" のように見えますが、よく見ると改善・改良の余地がたくさんある。完璧なものなどないのです。そして、それを改良するのは私たち自身です。その事実に気が付くと、人生が変わり始めます。Makerになるのです。

――エンジニアなどのプロフェッショナルが、アマチュアの精神を持ち続けるにはどうすればよいでしょうか。

プロフェッショナルは、自らの腕を磨きつつ、アマチュアのアイデアを寛大に受け入れること、そしてアマチュアを育てていくことが大切です。アマチュアが犯す "質の高い勘違い" は、時にイノベーションの源泉となります。アマチュアは、秩序をよい方に乱す存在なのです。プロは、そのようなアイデアをビジネスに取り入れるため、プロジェクトチームの中に、アマチュアを数名参加させるとよいでしょう。

独創的なアイデアを持つ人達には、町工場などで働くベテランエンジニア、伝統的な技能を持っているエキスパートが含まれます。こうした人達にも私は光を当てたいのです。どれだけ優れた力量を持っているか、本人はあまり気付いていない。周囲もこうしたエキスパートにあまり敬意を払わない。米国でスターリングエンジンという動力機関を再現している Maker のグループがあります。そこに参加するシニア層のエンジニアは、スーツを着て働くお父さん世代が、子供たちから頼まれても直せない類の機械の知識や製造技術を持っている。世代を超えた交流が新しい価値を生み出すことがあります。

日本にもすごい技術を持ったエキスパートがたくさんいますね。しかし、そういう方々の多くはホームページどころかメールアドレスさえ持っていない。その人の腕前を必要しているのに、すぐに見つけることができません。そこで、こうした能力を持った人がどこにいるかすぐに探せる人材マップを作れないかというアイデアを、城一裕さん（情報科学芸術大学院大学講師）らと相談しています。コミュニティやエリアをつなぎ直す取り組みになりそうです。

――Maker が社会を大きく変える時が来ているのでしょうか。

そうですね。ただ、分野によっては長期的な視点が必要です。というのも、いろいろな規制やルールに縛られている大人たちの考え方は、なかなか簡単には変わらないからです。既存の社会を相手に真正面から変革を訴えても頑なに阻まれてしまいます。でも、そういう時は、横から入るという手があります。

例えば、学校の生徒たちにものづくりを経験させたい、といっても「レーザーカッターを購入する予算がない」と一蹴されることがよくあります。しかし、学校の脇にある空スペースなどをちょっと拝借し、野菜を育て始めてみる。最初は誰も見向きもしない。しかし、そこにニンジンなどが芽吹いてくると、学校の生徒たちが興味を持って覗きにくる。みんな自分が口にするものがどこから来るのか興味があるからです。そのうち先生も覗きに来る。そうやって気付くとみんなを巻き込んでいる。道具や場所があるに超したことはないですが、何かも最初から用意する必要などないのです。

道具よりはるかに重要なのは、マインドを育てることです。そして正面突破をせず、横から入ってしまう。許可を求めず、小さく始めてしまう。Maker ムーブメントもそういう風にして拡がっています。こうしたコミュニティの中で育った子供達は、単なる Consumer ではなく、Maker として育っていきます。社会を改善できる人に自然となれるのです。子供たちが、消費者ではなく、文化の創造者である、と当たり前に思える社会が来たときに私は、世界が変わると確信しています。

――デールさんのいう Maker 精神を USP MAMGZINE も見習って、さらに今後も活動の輪を拡げていきたいと思います。今日は本当に有難うございました。

Maker space
～各国に広がる子供のための「出前工房」～

雑誌「Make」と Otherlab が連携して展開するワークショップ・プロジェクト。衣服の製作や木工などに必要な道具や材料を置いたスペースを設置し、子供たちに「作る」という体験を通じて、ものづくりやサイエンスに興味を持ってもらうのが狙いだ。参加費や材料費などは基本的に無料。米国の DARPA などが資金面で活動を援助している。

http://makerspace.com/

What is a maker?

Maker とは何か。それを分かりやすく示したものが、下記リストである。Maker に共通する 5 つの行動原則を示したものだ。Maker ムーブメントに参画する有志が考案したものだが、デール氏は好んでしばしばこれを引用している。

1）Interest in interdisciplinary projects
学術的な専門分野や組織の枠組みを超えた人々との出会いを楽しもう。クロスボーダーの交流が新たな刺激や気付きを生むからだ。

2）Desire to visualize and understand physical world
テクノロジーや情報を消費するだけでなく、自ら作り、参加せよ。作ることを通じて現実世界を可視化すれば、世の中をいっそう深く理解できる。

3）A mindset of playfulness
シリアスになり過ぎない。遊び心を忘れない。ツールセットより大事なのは、マインドセット。ないものは作れ。

4）Share in open source communities.
いつもオープンであれ。どんな研究開発でも、研究所、企業、大学のなかに留まっていては、ブレークスルーを成し遂げられない。異なるアングルから眺めた時に発見がある。

5）Sense that world can be improved, problems can be solved, and things can be changed.
自分を信じろ。スティーブ・ジョブズのように「世界は自分達の手でよりよく変えられる」「問題解決できる」という楽観的なモノの見方こそ最大の武器だ。

シェルスクリプト大喜利 第六回

#!$!#!$!#!$!#!$!#!$!#

司会：『もっと吹く』編集長・みかん

皆様こんにちは、はじめまして。三か月のご無沙汰、シェルスクリプト大喜利（略して sh 大喜利）のコーナーです。

ちょっと皆さん、聞いてくださいよ！このシェルスクリプト大喜利のコーナーの第四回で募集したちんじゅうちゃんマウスパッド。本当につくっちゃったんですよ。↓

マウスパッドを使う GUI 派な人に向けて、ちんじゅちゃんが CUI 文化の素晴らしさを教育するというお節介極まりないマウスパッド。しかし、**教育のために取り上げている sl コマンドネタは、肝心の GUI 派にとって、よく意味がわからないというオチが……。** ダメじゃん！

でも逆に、我々 CUI 派には大ウケで、8 月にコ○ケで頒布したら大半の方がニヤリとしてくれました。良いような悪いような……。でもまたこういう企画やりたいもんですなぁ。TechLION や OSC などのイベント入手できますよ。

周囲に CUI 文化の素晴らしさを説きたい人がいたら、こっそりマウスパッドをコレにすり替えちゃいましょう。

えーではそろそろ、本コーナーのシステムをご説明します！

シェルスクリプト大喜利とは

シェルスクリプト大喜利特有のルール

一、sh 大喜利はクイズやテストではありませぬ。なので決まった答えというものはないのです。あえて言うなら面白いスクリプトが正解！

二、面白いスクリプトとは例えば、こんなもの。

イ、人が考えつかない意外性がある

ロ、美しい or 芸術的 or 記述がシンプル・短い or 高速

ハ、アイデア・こだわりが光る

ニ、ネタになるようなバカバカしさ、くだらなさがある

などなど、ただし最後のは段位強制返還の恐れありよ。:-)

三、スクリプト動作環境は Linux とします。そして、特に断りなき場合は、Linux JM(http://linuxjm.sourceforge.jp/) に記載されているコマンド及び機能のみ使用可能とします。これは多くの人が楽しめるようにするためなのです。（但し JM に載っているので、**C シェル系での回答も OK！**）

四、sh 大喜利はシェルスクリプトを披露する場なので、**Perl や Ruby、Python などは使っちゃダメ**です。そもそも JM にも載っていません。逆にシェルスクリプトにとって不可欠な awk や sed 等は OK です。JM にもありますし。でも、よっぽど面白ければ、なきにしもあらず？

ルールもおさらいしたところで、それじゃイッてみよう。

五、Open usp Tukubai(http://uec.usp-lab.com/) も使用 OK ！但し、それなりに見応えないと採用はキビシいですよ〜。

本番開始

<第一問>
「幸運数」と呼ばれる自然数があります。100 以下の幸運数を全て求めるシェルスクリプトを書いてください。

皆さん知ってました？幸運数って数学的な定義があるってことを。ラッキー 7 ってのも何となく 7 なのではなくて、数学的に 7 が幸運数だからみたいですよ。知らなかった方は詳しくは Wikipedia 等で調べてもらうことにして、そうやって数学的に定義されているアルゴリズムをシェルスクリプトで実装してもらうのがこのお題というわけです。

投稿数で判断すると、このお題が今回の一番人気でした。個性豊かな解答が来とります。さてさて早速見ていきましょう。

◎Kさんの解答

```
1 #!/bin/sh
2 if [ $# -le 1 ]; then
3     N=${1:-100}
4     seq 1 $N | ./LuckyNumber.sh $N 1 2 | tr '¥n' ' '
5     echo
6     rm LuckyNumber.[0-9]*
7 elif [ $1 -lt $3 ]; then
8     exec cat
9 else
10     N=$(($1-$1/$3)); L=$(($2+1))
11     i=1; j=1
12     while read line; do
13         if [ $i -eq $3 ]; then
14             i=1
15         else
16             if [ $j -eq $L ]; then
17                 M=$line
18             fi
19             echo $line; i=$(($i+1)); j=$(($j+1))
20         fi
21     done > LuckyNumber.$$
22     ./LuckyNumber.sh $N $L $M < LuckyNumber.$$
23 fi
```

自身を再帰的に呼び出しているのでこのスクリプトはLuckyNumber.shという名前にしてください。

Kさんは2つ投稿してくれたのですが、私の独断と偏見でAWKを使ってないこちらを採用しました。添付コメントによれば、データを2回走査しなくてよい方法を思いつかず悩んだそうなのですが、アルゴリズム上多分2回走査は必要なんじゃないかと思いますよ。

さて、このコード。再帰呼び出しで実装してるのが他の投稿に無くてちょっと面白い。**一段授与**！これで二段だ。

◎東京awkerさんの解答

```
1 #! /bin/sh
2 awk "BEGIN{for(i=1;i<=100;i++){print i}}" > ln
3 i=2
4 n=$(awk "NR==$i{print;exit}" ln)
5 while [ -n "$n" ]; do
6     awk "NR%$n!=0" ln > ln1
7     n=$(awk "NR==$i{print;exit}" ln1)
8     i=$((i+1))
9     awk "1" ln1 > ln
10 done
11 awk "1" ln
```

AWKに拘って、呼び出している外部コマンドをAWKだけにしたとのことです。なるほど……。

その拘りが素晴らしい。コードもそつなく書けてるし。よし**初段授与**だ！

◎sed内kerさんの解答

```
1 #! /bin/bash
2 rm -rf ln; mkdir ln; cd ln
3 echo 0{0..9}{0..9} | sed 's/000/100/' | xargs tou
  ch
4 a=2
5 b=`ls | sed -n "${a}p"`
6 while [ -n "$b" ]; do
7   ls | sed -n "${b}~${b}p" | xargs rm -f
8   b=`ls | sed -n "${a}p"`
9   a=$((a+1))
10 done
11 ls | sed 's/^0*//'
```

sedに拘ってAWKを一切使わず組んだとのことです。また、テキストファイルではなくファイルシステムを使っているところも拘りです、と添付コメントに書いてありました。

うーん、面白い。sedとファイルシステムを使いこなしてるねー、sedは一部GNU拡張な使い方してるけど。最初の1～100の値をブレース展開で作ってるのもいいね。よし、これは**二段授与**しちゃおうか。……だけど今紹介したこの二人、**もしや同一人物なのでは???**コードが似てるしペンネームも。言ってくれればまとめて三段にしてもいいからね。

そして、Tukubai使った解答は来ないのかーと、思っていたら一つ採用しようか悩む解答が来ましたよ。（結局採用）

◎312516さんの解答

```
1 #! /bin/sh
2
3 seq 1 100 > num
4 step=2
5 val=$(yarr num | self $step)
6 while [ $(gyo num) -ge $val ]; do
7   cat num     |
8   yarr -$val  |
9   awk 'maxnf<=NF{maxnf=NF;print;next;}{str="";del=
  "";for(i=1;i<=NF;i++){str=str sprintf("%s%s",del,$
  i);del=FS;}for(;i<=maxnf;i++){str=str sprintf("%s%
  s",del,"*");}print str;}' |
10  delf $val   |
11  tarr        |
12  grep -v '*' > tmp-num
13  val=$(yarr tmp-num | self $step)
14  step=$((step+1))
15  mv tmp-num num
16 done
17 exec cat num
```

Tukubaiらしく、綺麗にパイプで繋いでるなーと思ったら途中にごちゃごちゃしたAWKが……。これが残念です。

しかし投稿者コメントによると、「delfコマンドは列数が揃っていないと正しく動かず、やむを得ず列数を揃えるためのawkを入れました」とのことで本人も残念がってました。

うーむ、これは仕様ですね。tateyokoコマンドもそうなのですが、列数が揃ってないものはエラーとして弾きます。Tukubaiでは不正なデータは、それも受け入れられるようにコマンドがいちいち配慮せず、容赦なく弾きます。そうする事でコマンドをシンプルに設計するという思想だそうです。必要なら前段階で整形しなさいと。確かに一理あります。

ということで、これは仕様なんだけどもよくぞTukubaiを使ってくれた。ありがとう、**初段授与**。

長かったですけど、次のお題行きます。

<第二問>
9行9列、各列がスペース区切の九九のテキスト表ファイルがあります。この表を標準出力から読み、x行目、y列目にあるセルを含め、その下のセル全てを、上にz行詰めるシェルスクリプトを書いてください。ただし次の仕様とします。
- **詰めたことでできた空セルの値は"@"という文字にする。**
- **x,y,zの値は全てコマンド引数で渡ってくるものとする。**
- **1≦x,y≦9および、0≦z<xはチェック不要とする。**

Excelなんかでもよくありますよね。指定した場所のセルを消して1セル分上に詰めたり、あるいは一つ増やしたいから1セル分下にずらすとか……。そうやって行を跨いだ操作っていうのはテキスト処理としては面倒なはず！と思ってお題にしたわけです。

寄せられた解答はどれもTukubaiを使ってます。やはりこういうシーンこそTukubaiの得意分野というわけですな。

◎321516さんの解答

```
1 #! /bin/sh
2
3 sub=´´
4 i=1
5 while [ $i -lt $(($1-$3)) ];do
6   sub="$sub ¥¥$i"
7   i=$((i+1))
8 done
9 while [ $i -le $((9-$3)) ];do
10   sub="$sub ¥¥$(($i+$3))"
11   i=$((i+1))
12 done
13 while [ $i -le 9 ];do
14   sub="$sub @"
15   i=$((i+1))
16 done
17 sub=${sub# }
18
19 tateyoko |
20 sed $2´s/¥([^ ]¥{1,¥}¥) ¥{1,¥}¥([^ ]¥{1,¥}¥) ¥{1,¥
   }¥([^ ]¥{1,¥}¥) ¥{1,¥}¥([^ ]¥{1,¥}¥) ¥{1,¥}¥([^ ]¥
   {1,¥}¥) ¥{1,¥}¥([^ ]¥{1,¥}¥) ¥{1,¥}¥([^ ]¥{1,¥}¥)
   ¥{1,¥}¥([^ ]¥{1,¥}¥) ¥{1,¥}¥([^ ]¥{1,¥}¥)/´"$sub"´
   /´ |
21 tateyoko |
22 keta
```

先程に続きありがとうございます。

いやぁ、これまた途中にゴニョゴニョした行があるけどアイデアは面白い。やりにくい特定列の行詰めを回避するためにtateyokoで表を一時的に転置し、特定行の列詰めにしてしまっている。そこでsedを活用しているというわけか。これはいいね。**一段授与**！さっきのと合わせて二段だ。

◎gori.shさんの回答

```
1 #!/bin/bash
2 x=$1
3 y=$2
4 z=$3
5 self 1/NF > $$-kuku
6 cat $$-kuku | self $y | sed $(($x - $z)),$(($x - 1)
  )d | ycat $$-kuku - | head -n $((9 - $z)) | self $y
  NF | sort | cjoin2 -d@ key=$y - $$-kuku | delf $y |
  keta
7 rm $$-kuku
```

はい、Tukubaiリリース元、USPの中の人からの投稿です。コメント「tukubai使用でawk封印。がんばってパイプ一本にまとめてみました」、ということです。

さすが中の人、短くまとまってるな。なるほど、加工対象の行だけ取り出して切り詰め、インデックス付けて、元のデータとJOINするのか。お手本ありがとう。**一段授与**！えーと……これで五段。順調に段位を重ねてますな。

おまちどうさま。今回のおバカお題、第三問行きましょう。

<第三問>
catコマンドの作者が遺言を残しました。
「私の人生における最大の後悔は、dogコマンドを作り忘れてしまったことだ。あぁ、誰か作ってくれ。ガクッ」
さぁ！一体どんなコマンド？

作者を勝手に殺すな!とツッコミ入れられそうですが、そこはご愛嬌。飽くまでお遊びですから～。というわけでこれは、dogコマンドがもし実在していたらどんなコマンドかというのをシェルスクリプトで表現してもらうお題です。

力作来ましたよ。まず、その力作を紹介しましょう。

◎@bsdhackさんの回答

```
1 #!/bin/sh
2
3 curl="-s -o -"
4 wget="-q -O -"
5 lynx="-source"
6
7 # get command name and path
8 for i in wget curl lynx
9 do
10   for j in /bin /usr/bin /usr/local/bin /opt/bin /
  opt/local/bin
11   do
12     if [ -x ${j}/${i} ]
13     then
14       command="${j}/${i} `eval echo ´$´${i}`"
15       break 2
16     fi
17   done
18 done
19
20 # args
21 while [ ${1} != "" ]
22 do
23   case ${1} in
24     -- )
25       shift
```

```
26          break
27          ;;
28      -* )
29          opt="${opt} $1"
30          ;;
31      *  )
32          break
33          ;;
34    esac
35    shift
36  done
37
38  for i in $*
39  do
40    if expr ${i} : "https*://" > /dev/null
41    then
42      if [ -n "${command}" ]
43      then
44        ${command} ${i} | cat ${opt}
45      else
46        echo "${0##*/}: cannot get remote file: ${i}
    " 1>&2
47      fi
48    elif expr ${i} : "ftp*://" > /dev/null
49    then
50      case ${command} in
51        *wget*|*curl* )
52          ${command} ${i} | cat ${opt}
53          ;;
54        * )
55          echo "${0##*/}: cannot get remote file: ${
    i}" 1>&2
56          ;;
57      esac
58    else
59      cat ${opt} ${i}
60    fi
61  done
```

これ力作なのです！誌面が残り少なくて載せられないのが残念ですが、man ドキュメントまで添えられていたのです。

それでどういうコマンドかというと、cat はファイルを表示するのみだったところを、これは http/https/ftp にまで拡張しているんですね。more を拡張して less が作られたように、cat を拡張する方向で dog を作ったというわけですね。

man まで作る芸の細かさに感服した！その man を紹介できなくてすまない。でもその努力はちゃんと段位に反映させてもらうよ。ということで、**二段授与**！これで四段だ！

◎猿二号さんの解答

```
1 #! /bin/sh
2
3 url="http://usokomaker.com/weekly/?a=Maker&oo=$1"
4 curl -s "$url"                                   |
5 nkf -w                                           |
6 sed -n '/^<div class="maga_copy">/,/^<¥/div>/p'  |
7 sed 's/<[^>]¥{1,¥}>//g'
```

この解答、curl とか nkf とか使ってるんですけど**面白いから許します。**curl、nkf コマンドが無い方は是非インストールしてお楽しみ下さい。あと、端末表示は UTF-8 にして下さい。

さてさて、これは一体何なのか？添えられていたコメントを読んでみると……「cat が con**cat**icate の略なら、dog はきっと pseu**dog**raph（＝ウソ文書）の略なはずです。なので引数文字列を元にウソ文書を吐きます。笑ってください」とのことです。

これは……、100% くだらない文章を吐くことで有名な**うそこメーカーじゃないか!**アタクシもこのサイト好きだけどね。それで、どのメーカーを使ってるのかと思ったら「週刊誌メーカー」かい！東○ポもびっくりゴシップ記事というわけね。うそこメーカーさんにも感謝しつつ、**一段授与**。これで二段だ。

というところで本日の大喜利はこれにてお開き！読者の皆さん、投稿してくれた皆さん、ありがとうございました。

投稿大募集!!

次回のお題

一、1 ～ n（n は引数で指定）までに存在する「完全数」を列挙するプログラムを書いてください。速さを追求した解答、または Tukubai を使う等して (使わなくても可) コードの美しさや面白さを追求した解答を待ってます。

二、URL エンコーダー（URL 化できるよう一部の文字を %xx にするアレ）を作ってください。勿論 Perl とかそーいうのはナシですよ。使っていいのは Bash,AWK,sed,tr 等、Linux JM に載っているシェルとコマンドだけです。

三、"a" という名のテキストファイルがあります。a の内容を表示するやり方 (一番簡単なのは cat a でしょう) を沢山列挙してください。列挙する数で勝負するのでも、珍しいやり方を列挙して勝負するのでも OK。

投稿のしかた

お題への回答は、お名前（ペンネーム）、回答したいお題番号、回答スクリプト、簡単な補足の四点セットで下記の宛先へ！一人何問でも何個でも回答可です。尚、次回締め切りは **11 月 26 日（月）午前 0 時**とします。しかもその間は何度でも回答の修正を受け付けます。

お題もどしどし送ってくださーい

それからお題も大募集。考えてくれた方にも段位を授与します。自分で出題して回答するのも、今のところ可！

投稿先

どちらも投稿先は、mag@usp-lab.com です。面白ければ（ワリと）何でもありなので、どしどし投稿待ってます！

数独（Number Place Puzzle：ナンプレ）は、クロスワードパズルのように、行列の中にある空マスを、正しい数字で埋めていくクイズです。

第３回

こんにちは、レイコです。
秋の夜長も Tech な数独よ！

数独クイズ作者：gori.sh

		3	4	8	9	1		
	5	4	2	3	1	9	6	
	9						2	
	1							
		5	1	9	2	3		
							9	
	3						1	
	4	2	8	1	5	6	3	
		8	9	7	3	1		

「数独 (SUDOKU)」はニコリの登録商標です。

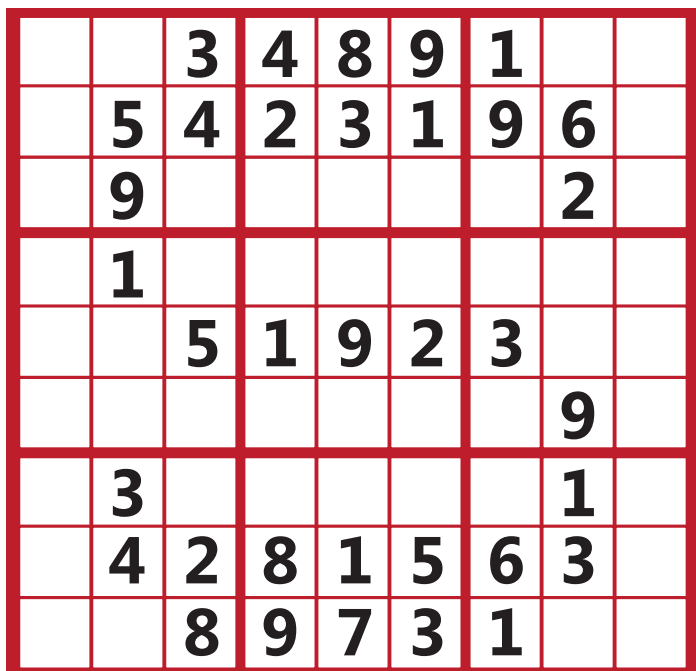
遊び方

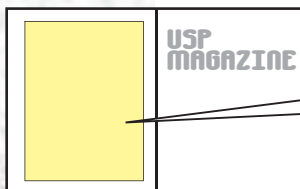
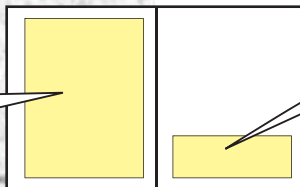
（1）使ってよい数字は、1から始まる自然数。

最も大きい数字は、一行（一列、一つのサブブロック）に入れられる上限値です。

例えば、「9 × 9」マスのパズルであれば、使ってよい数字は「1」から「9」までです。

（2）一つの行に、各数字は一度しか使えません。

（3）同じ列、同じサブブロックのマスを埋める際も、各数字は一度しか使えません。

前回の解答

9	7	8	4	1	5	2	6	3
2	4	5	6	8	3	1	7	9
1	3	6	7	2	9	8	4	5
8	5	7	9	4	2	6	3	1
3	6	1	5	7	8	4	9	2
4	9	2	3	6	1	5	8	7
7	8	3	2	5	4	9	1	6
5	1	9	8	3	6	7	2	4
6	2	4	1	9	7	3	5	8

天地概況②　―奈須螢路―

二〇〇一年九月一一日、米国の通称「ツインタワー」に、ハイジャックされた二機の旅客機が体当たりし崩壊しました。私は少し前にMITにある用件で赴いており、もし日程がずれていたらとテロに恐怖しました。数日後、西氏（アスキー・MS元副社長）が一枚の写真を、「凄いだろ、こういうの見れたらイイだろ」と手渡してきました。それは後に「グランドゼロ」と呼ばれた、タワーの崩壊地点を遥か上空から撮った衛星写真と思われるものでした。当時インターネットでも航空写真提供サービスはまだ存在せず、リアルタイムの衛星写真を見ることは一般には無理で軍事機密でした。しかし私はこの時写真を見せた意図に全くピンと来ず、むしろ冷戦後続くテロに「21世紀早々にコレでは、この先一体どうなる?」と思うだけでした。

それから数年、急成長した検索サービスのグーグルはグーグルアース（ストリートビュー）をリリース。リアルタイム映像でこそ無いものの、「グランドゼロ」のその後を見る事ができました。しかしグーグルの映像提供はプライバシーの問題と利便性とから賛否両論を巻き起こす事になります。実際、我が家は布団が干された少々カッコ悪い写真が衆人監視のネット上に掲載されたのです。

一方で初めて行く場所やお店の様子が分かるなど利便性は計り知れません。人々はこのメリットとデメリットの中でプライバシーの侵害等、そのリスクにある種の覚悟を強いられる事になります。

近年では実名主義のSNS、フェイスブックが既存の匿名主義のSNSに代わり世界の主流となり、人類の九人に一人が利用する状況となりました。新しい出会いを生み、旧友を再会させ、知人とのリアルタイムな近況交換を促進しました。企業もこれを活用、更に政治の場にも拡りをみせています。

これらのツールはプライバシーの侵害という大問題を解決できずにその活用の場を爆発的に広げてきました。これに苦言を呈す人も多く、被害者を多数生んでいます。しかし私の印象では実際には両社ともこの問題を解決する気が無いように見えます。むしろプライバシーという概念の破壊こそが両者の真の目的かとさえ思えてきます。

一方、未だインターネットを自由に利用できない国が多く存在します。そんな国でも人々は徐々に密かにネットで交流を計っています。もしグーグルが撮り貯めた写真ではなく、リアルタイム映像を提供したらどうなるでしょう。一部の権力者やテロリストが機密情報に触れる危険もある一方で、全人類がテロの本拠地を探し権力の中枢を監視し晒してしまうやもしれません。軍事力は全貌が秘密だからこそ相互抑止力を発揮します。これまでは敵対する双方の同時軍縮は実際には無理でした。しかしネットは双方の軍事機密を暴くことで無意味化させる力があります。かつて西氏が私に問うた、一部の権力者が独占する技術（＝衛星からの地上監視）の一般開放は既に技術的には可能になり、活用が期待されます（実行はまだ無理か?）。

「他者を理解する為に情報を得、自分の情報を公開する」。知らない事が恐怖を生み対立するのだという確信が両サービスの根本思想にあるのではないか、と私は想像します。つまり「守秘義務による安全で安心」ではなく「秘密など無いから安全で安心」なのです。この新しい常識がプライバシーという概念の破壊から始まる情報共有と相互理解により構築されるのです。すると世界平和の実現に対する最強最後の抵抗勢力は、独裁者や権力者や巨大資本家では無く、実は一般人の守秘願望なのかもしれません。でも実際は中々進まないだろうなぁ。秘密の共有って魅力的だし。ああ、どうやら私は旧人類なのかも……。

【著者プロフィール・なすけいじ】

外部役員、監査、顧問として各計画に横断的に参加する。作業員としてずっと現場戦力でありたい。

USP MAGAZINE 2012 autumn

(vol.6)

編集長　松浦智之
編集　鎌田広子
青井大地
ライター　柏崎吉一
制作協力　USP 友の会
表紙デザイン　石塚幸治
（ジーズバンク）

発行人　當仲寛哲
発行元　（有）USP 研究所
定価 400 円（本体価格 381 円）

ご意見・ご感想・投稿はこちら
mag@usp-lab.com
（技術的なお問合せには対応しかねますので
ご了承ください）

USP Magazine info

USPMagazine の細かい情報が定期的に
アップアップデートされています。
訂正のおわびもこちらから ----->>

USP マガジン FB

編集後記

・同人作家からスタートし、私が最初に作ったのはオライリーのオマージュ本だった。でも内容は完全オリジナル、表紙イラストも自分で描いた。そんなことしてきた私が活動 10 年目にしてその創業者に会うことになろうとは！本を見せたら「ワーォ！」と嬉しい一言、Maker 魂が伝わったはずだ。でも側にいらした社長さんから「弁護士に相談します」とコワいジョークが！（松浦）

・Turing さんのイラストを描こうと思って、写真から緻密に描写しようと思ったんですけど、描いてみると普通に整った顔なので面白くない。（禿おじさんなどは描いてて楽しい）なので髪型に特徴を持たせて愛嬌ある感じにしてみました。モジリアーニみたいに、変顔でイラスト芸術を追及してみようかなぁ。（鎌田）

・紙の上にペンとインクで描く、写真を撮る、コンピューター上で作図してレイアウトする、キーボードから文章やプログラムコードを打つ……どの作業も、自分の魂を投影する作業。魂を成長させることと、それを思うように投影する能力を身につけること。「ものづくり」に必要なのは、終わりのない精進！（青井）

・盆休みに宮城県石巻市内の宿に一泊した。夜、近くの居酒屋に入ると多くの客で賑わっていた。市街は活気を取り戻しつつあるようだった。波にさらわれた海辺や、学校の校庭には街から集められたガレキがいまも山積みである。この一年半、立て直しに費やした労苦は想像もつかない。せめて体験を語り継ごう。そして地元で獲れた旨い魚と酒をじっくり味わおう。（ライター 柏崎＠エクリュ）

次号予告

2012 年 12 月発行予定です。

- Tukubai on FreeBSD
- TechLION 再録　村井純教授
- ○○に訊く技術者哲学
- 今私たちは何を学ぶべきか
- USP 友の会　納涼船吉報
- ほか　好評連載陣

内容は変更になる場合があります。

TALK LIVE

FRIDAY 16 NOVEMBER
TechLION vol.10
むらいじゅん

MC：法林浩之、馮富久　TalkGuest：ほか交渉中

2012.11.16 Friday　Place：六本木 SuperDeluxe

Start 19:30 ～ Close 22:00（予定）　Ticket：前売り・予約　2700 円、当日 3200 円（1ドリンク込）

主催：ユニバーサル・シェル・プログラミング研究所　協賛：技術評論社、LPI-Japan、ヌーラボ、他

後援：日本 UNIX ユーザ会、EMZERO、キャリアデザインセンター「エンジニア Type」

新感覚の技術トークライブ「TechLION（てっくらいおん）」 詳細は WEB で！　TechLION vol.10　検索

TechLION サイトはこちら♪＞　http://techlion.jp

USP 友の会　会員募集中

シェルでつながる技術の輪！基礎から、プロのライブコーディング・ライブまで、いろいろやっています。おもいっきり真面目に、時に面白く、心から楽しめます。WEB 会費は無料です。勉強会や定例会の参加費は営利目的ではない為、良心的です。

UNIX/Linux/ シェルスクリプトの可能性を極限まで追求します！

詳しくは

http://www.usptomonokai.jp にアクセス！

Facebook もやってるよ♪＞

USP 友の会

USP MAGAZINE 入手方法は？

vol.0　2011　spring

vol.1　2011　summer

vol.2　2011　autumn

vol.3　2012　winter

vol.4　2012　spring

vol.5　2012　summer

自炊不要　定期購読又はバックナンバーをお申込みいただくと、PDF 版が無料で手に入ります。

大手書店でも売ってるよ♪＞　http://www.usp-lab.com/pub.store.html

定期購読申込はこちら♪＞　http://www.usp-lab.com/pub.order.html

本体は、定価 400 円（本体価格 381 円）